University of Windsor

Scholarship at UWindsor

Electronic Theses and Dissertations

Theses, Dissertations, and Major Papers

2019

Building Teams of Experts using Integer Linear Programming

Sagarika Narendra Khandelwal
University of Windsor

BUILDING A TEAM OF EXPERTS USING INTEGER LINEAR PROGRAMMING (ILP)

By

Sagarika Khandelwal

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

2019

BUILDING A TEAM OF EXPERTS USING INTEGER LINEAR PROGRAMMING (ILP)

by

Sagarika Khandelwal

APPROVED BY:

J. Wu
Department of Electrical and Computer Engineering

A. Mukhopadhyay
School of Computer Science

J. Chen, Advisor

School of Computer Science

May 10th 2019

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

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

ABSTRACT

Given a set of projects, each requiring a set of specific skills, and given a set of experts, each possessing a set of specific skills, the cluster hire in a network of experts seeks to find a suitable subset of the experts to jointly accomplish a subset of the given projects with their complementary expertise. We consider the problem of selecting an optimal team of the experts in terms of maximizing the profit that the selected team is able to generate, where the profit is determined partly by the revenue of the projects this team is able to accomplish, partly by the efficiency of the team measured by the prior collaboration experience among its team members. This optimization is further constrained by the given workload capacity of each expert, and by a given budget on team hiring. We approach the optimal solution with Integer Linear Programming (ILP) technique and compare its result with those from other heuristic solutions.

DEDICATION

Dedicated to Lord Krishna, my baba Satyanarayan Khandelwal, my dear papa Narendra Khandelwal, my beloved mumma Namrata Khandelwal, my beautiful sisters Komal & Anjali, my adorable nephew Vivaan

And also, the rest of my family and friends

ACKNOWLEDGEMENTS

I owe a debt of gratitude to Dr. Chen, for the vision and foresight which inspired me to conceive the thesis work. As my teacher and mentor, she has taught me more that I could ever give her credit for here. I am thankful to my thesis committee members, Dr. Mukhopadhyay and Dr. Wu for providing me extensive personal and professional guidance which help me lean a great deal about both scientific research and life in general.

Nobody has been more important me in the pursuit of this thesis than the members of my family & friends. I would like to thank my parents & Dhruv, 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 School of Computer Science and my friends who provided unending support and encouragement through my course work at University of Windsor.

TABLE OF CONTENTS.

DECLARATION OF ORIGINALITY iii

ABSTRACT iv

DEDICATION v

ACKNOWLEDGEMENTS vi

LIST OF TABLES x

LIST OF FIGURES xi

CHAPTER 1 INTRODUCTION 1

1.1 Overview 1

1.2 Motivation 3

1.3 Problem and Solution Outline 3

1.4 The scope of the thesis 5

1.5 Structure of the thesis 6

CHAPTER 2 BACKGROUND STUDY 7

2.1 Overview of the research 7

2.1.1 Data Mining 7

2.1.2 Team formation 8

2.1.3 Cluster Hire 8

2.2 Fundamental Concepts 11

2.2.1 Linear Programming 11

2.2.2 Integer Linear Programming(ILP) 12

2.3 Application of Linear Programming 13

2.3.1 Example of Integer Linear Programming 15

2.4 Algorithm 17

2.4.1 Exact Algorithm 18

2.4.2 Heuristic Algorithms 20

2.5 Literature Review 21

2.5.1 Team formation Problem and cost functions for best team formation in Social Networks.......21

2.5.2 Social Network an important factor for efficient team formation.......22

2.5.3 Genetic Algorithm for team formation.......22

2.5.4 Team formation problem solved using various techniques.......23

2.5.5 Clustering Technique for Team Formation Problem.......24

2.5.6 Online Team formation in Social Networks.......24

2.5.7 Cluster Hiring Problem.......25

2.6 Problem Statement.......26

CHAPTER 3 IMPLEMENTATION DETAILS.......30

3.1 Notations used in ILP.......30

3.2 ILP model for maximized profit.......31

3.2.1 Justification of the ILP.......33

3.3 ILP model with maximum profit and minimum communication cost.......34

3.4 Heuristic Algorithm for Cluster Hire Problem.......36

3.4.1 Cluster Hire Algorithm for maximized profit and collaborative team.......37

3.4.2 Sample data set observations.......40

CHAPTER 4 RESULT AND ANALYSIS.......43

4.1 Heuristic vs Cluster Hire Profit Model (10,000 Nodes).......47

4.1.1 Total Profit vs Budget.......47

4.1.2 Number of Projects completed vs Budget.......49

4.2 Heuristic vs Cluster Hire Profit and Communication Cost Model (50 Nodes)...50

4.2.1 Total Communication Cost vs Budget.......50

4.2.2 Total Profit vs Budget.......51

4.3 Discussion.......52

CHAPTER 5 **CONCLUSION AND FUTURE WORK**.......56

5.1 Conclusion.......56

5.2 Future Work.......56

REFERENCES/BIBLIOGRAPHY.......58

APPENDICES.......60

Appendix A 60

VITA AUCTORIS 63

LIST OF TABLES

Table 1 Notations in ILP 31

Table 2 Expert Attributes 41

Table 3 Project Attributes 41

Table 4 Variable Size 46

Table 5 Scale of values used 47

LIST OF FIGURES

Figure 1 Data Mining 8
Figure 2 Cluster Hire 10
Figure 3 ILP using Branch and Cut solver 20
Figure 4 Social Network of Experts 27
Figure 5 Heuristic Algorithm for Cluster Hire Problem 39
Figure 6 Sample Network of Experts 40
Figure 7 Comparison between budge and profit with project size 20 48
Figure 8 Comparison between budge and profit with project size 10 48
Figure 9 Number of completed projects vs budget (10 projects) 49
Figure 10 Number of completed projects vs budget (20 projects) 49
Figure 11 Communication cost vs Budget (08 Projects) 50
Figure 12 Communication cost vs Budget (05 Projects) 51
Figure 13 Profit vs Budget (05 Projects) 51
Figure 14 Profit vs Budget (08 Projects) 52
Figure 15 Problem Size Vs Memory Size 54
Figure 16 Problem Size VS Execution Time 55

CHAPTER 1
INTRODUCTION

1.1 Overview

One of the most common problems studied in today's world is that of efficient resource allocation and scheduling. Discovering a team of experts through a recruitment process, the recruiter seeks for a team that can accomplish the tasks within a deadline and the team so hired is cost-effective.

A project is a well-designed and planned approach to collaborate with individuals to achieve organizational aims and company targets. Each project consists of various task and activities that need experts who have deep expertise and skill necessary to perform that task. Organizations need such a team of experts who possess the right skills for the smooth delivery of the project. This problem where we seek to discover a team of experts such that all the skills required for a project are covered is called Team Formation Problem(TFP)

Cluster Hiring Problem involves the hiring of individuals for a given set of projects such that the hired experts hold all the skills required by this set of projects. Also since each expert has its own cost (i.e. salary) we make sure we hire individuals within a budget allocated for this set of projects. The CLUSTER HIRE problem was first introduced by Golshan et al. [13], where they provide an overview of the problem through the online labor market websites like Freelancer.com and Guru.com

The team formation involves hiring groups of experts in big institutions and hiring organizations where each expert holds its complementary skills which help with effective problem solving and successful delivery of the project. In other words, given a set of experts each having a diversified set of skills, the goal is to hire a team of experts that can collectively cover the required skills of the experts, making sure that the hired group of experts make the most collaborative team. When the project is completed each organization obtains a profit. Each set project has its own set of skills, and the number of experts needed per skill. Hence the set of experts hired should possess all the skills required for the set of projects. For recruiting each expert, the organization has to pay the expert an economic cost (i.e., the salary of the expert). The organization is also assigned a budget which cannot be exceeded in the process of hiring the group of experts. In other words, the sum of the salary of the individuals hired for the set of the project should not exceed the company’s defined budget. Such setting helps in maximizing the profit of the organization.

Grouping experts who have worked together in previous projects, helps in faster completion of a project as well as in producing qualitative outcomes. Assuming that there exists a communication channel among the experts, communication cost between two experts can be calculated based on the prior work experience. If any two experts have previously worked together in the past, then it is believed that they have a better frequency of communication compared to experts who are introduced in the team for the first time. Having a good frequency provides many benefits like the quality of work and effectively meeting the deadline of the projects. Also, grouping individuals with minimal or no communication between them can increase the expense of the project thereby decreasing the revenue obtained from the project. Determination of the communication cost depends on the need of the project. Eventually, organizations look out for teams that give more profit to the projects.

Working capacity of an expert is another important parameter while hiring an expert for multiple projects. The recruiting managers also make sure that the employee’s capacity is not exhausted in the process of hiring experts that generated the most revenue for the organization. Each expert is associated with maximum capacity he/she can offer. The combination of communication cost between two experts and the capacity of each expert helps in providing multiple feasible teams from which a team that produces the maximum profit can be obtained.

For solving Cluster Hire problem, we introduce features such as the capacity of an expert, collaboration cost between two experts, number of experts demanded by each skill per project, budget assigned for hiring experts for a set of project and profit gained by each project for successful completion of a project.The features are explained in details in the later part of the thesis.

1.2 Motivation

In the immersing world of technology, many companies are growing worldwide where the owners need to select a team of experts with the collective expertise required to benefit from various opportunities that have been identified within the market that the company targets [13]. In this context, it is important to make efficient resource allocation so that certain objectives of the organization,such that meeting the organizations' lawful and social commmitments with respect to the systhesis of its workforce can be achieved.

Another relevant motivation comes from the online labor market websites such as Freelancer and Guru where they get multiple projects from multiple remote locations. At first, the freelancers work independently but later with the increase in project size and complexity; these experts collaborate to form a consulting company which receives projects, skills and the number of individuals required for the projects. For each skill required for a project, the consulting company hires experts to fulfill the set of demanded tasks. There is a budget defined by the financial department for hiring experts for the project and hence the total salary of all the experts should be less than or equal to the budget. The goal of the company is to earn the most profit by completion of the projects.

To overcome the above-described problem, people have proposed many ways of discovering the most collaborative team for a single project and a profit-maximizing team for a set of projects which consists of the most collaborative experts

1.3 Problem and Solution Outline

Cluster Hire problem has been researched by many to form a group of experts having the skills required to complete a subset of projects producing maximum profit under a defined budget. A variation of this problem is Cluster hiring in a social network, where along with covering a subset of projects the experts are selected on the basis of their prior collaboration with other experts in the group such the experts selected have good compatibility and better communication which helps in completing the project in a timely and qualitative manner.

We study the cluster hire problem, with the following assumptions

1. Each expert possesses a set of skills.
2. Experts have workload capacity indicating how many maximum number of skills can be assigned.
3. Each expert has an associated hiring cost associated. The hiring cost is calculated only once. That is, an expert can be assigned to a number of projects and can perform a number of skills based on his/her working capacity.
4. There is a set of projects, each having certain skills and the number of experts having that skill needed to complete the project.
5. The social network amongst the experts, retrieved from previous work experience.

The experts are connected in a network which is modeled as a graph. Each expert is associated with a node of the graph. An edge connects two experts in the graph with edge weight representing the strength of collaboration between two experts obtained from prior collaboration in the past. When the experts are not connected, the weight is assigned by the sum of weight on the shortest path between the two experts. The least the communication cost the most collaboration among the experts. For example, if there are two pairs of experts where the first pair of experts have worked in ten projects in the past, and the other pair of experts have worked in nine projects in the past, then the former experts will be selected since they tend to have better frequency i.e communication cost compared to the latter pair of experts. We model the communication cost among the team members using the sum of distances between each pair of experts in the group.

For completing a given set of projects, we are provided with a predetermined budget to be spent on hiring experts. We first solve the problem of hiring the group of experts for a set of projects considering the team to be the most collaborative and producing the maximum profit.

We propose to solve this problem using mathematical programming for linear problem solving. Chapter 2 explains in detail the problem statement and our approach.

1.4 The scope of the thesis

Kargar et al. [2] study the team formation problem, which is a subclass of Cluster Hiring problem. They propose a new formulation to calculate the communication cost between two experts called the *sum of distance* function. In their study, they all provide a detailed analysis on the computational complexity, approximation of the problem solution to the optimal solution as well as heuristic solutions. They demonstrate the efficacy of their approach by carrying out experiments on real datasets of experts and demonstrated their advantages and baselines.

Meet et al.[14] first studied cluster Hiring problem in a network, where they aim at hiring a group of experts for a set of projects which produces the maximum profit in which each project has an associated revenue, amount associated, which is obtained when the project completes. In their paper, they also study various attributes associated with a team and its members such as collaboration between two experts, the capacity of an individual expert, a salary of the experts. They also make sure that the hiring cost of the experts is within the budget assigned by the financial department of the company. They further propose two greedy algorithms and their results were obtained through experiments done on a large graph of experts.

We propose to solve the cluster hiring problem with a slightly extension to the one proposed in the work. In our setting, we have a set of projects, each having skills and the number of experts required per skill for project completion. Our aim here is to maximize the profit earned by the summation of the revenue associated with each project. Furthermore, we modify our proposed solution so that it produces maximized profit and also selects experts that have the most collaboration among each other.

We apply Integer Linear Programming (ILP), which uses mathematical programming for solving complex decision-making problems. We show the ILP models created to solve the cluster hiring problem with and without social network and further compare our work with a heuristic solution.

1.5 Structure of the thesis

The remaining thesis is organized as followed. In chapter 2, we review the basic concepts of Data Mining. Team Formation and Cluster Hire. This chapter also involves a detailed explanation of the problem statement of the thesis, review of the ILP and greedy approach used for the solution. We then present our approach in Chapter 3 which includes detailed explanation of the ILP models, its objectives and constraints along with the Heuristic greedy algorithm designed for the problem Chapter 4 presents the contribution of this thesis work and describes the implementation details of the approach and the results showing the comparisons of the approaches. Chapter 5 provides the summary concluding the thesis along with the direction for possible future work.

CHAPTER 2
BACKGROUND STUDY

2.1 Overview of the research

2.1.1 Data Mining

The process of discovering hidden patterns in huge data sets using methods like machine learning, statistics, and database systems is called Data Mining. It is one of the subfields of computer science and statistics with an overall goal to extract information from data set and transform the information into a well-described structure for further use. Data mining helps in extracting important insights from the data; this is commonly known as Knowledge Data Discovery. To keep up with the pace of data generation, it is needed to have good data mining technique to discover interesting patterns to analyze the trend. Depending on these trends an organization might enhance the decision-making process to generate more revenue or attract more customer for its business. Applying such techniques are costly and may not be affordable to every type of business. People choose different techniques based on the nature of the data and that is what determined the fate of these companies.

Knowledge Discovery in Database is done in step by step procedure: [6]

- Data Cleaning: In this step, the noise, and inconsistent data are removed.
- Data Integration: In this step, multiple data sources are combined.
- Data Selection: In this step, the data relevant to the analysis task are retrieved from the database.
- Data Transformation: In this step, data is transformed or consolidated into forms appropriate for mining by performing summary or aggregation operations.
- Data Mining: In this step, intelligent methods are applied to extract data patterns.
- Pattern Evaluation: In this step, data patterns are evaluated.
- Knowledge Presentation: In this step, knowledge is represented.

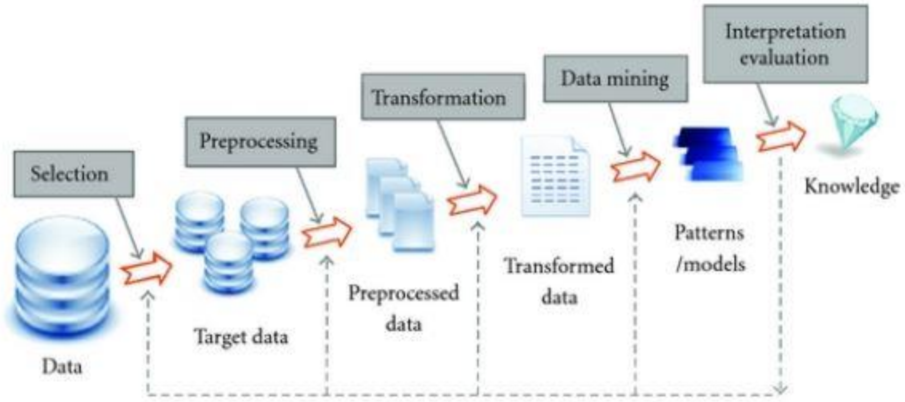

Figure 1 Data Mining

2.1.2 Team formation

The concept of team formation is to hire a set of experts who hold expertise in one or more skills that is needed for completion of a project, which will give benefit to an institution upon its completion. This helps organizations to complete projects undertaken and also helps experts strengthen their skills and position in the company by performing well. The main idea behind this concept is to gather individuals having expertise in the skills required for successful completion of the project. Selection of individuals to work in a team can be made in various ways using different constraints and factors. For instance, a team benefits from the similarities in the background among the team members. This can help decrease any conflicts or miscommunication. Having few differences can also help reduce the amount of time it takes to become an effective working group since it is less needed to adjust individual working styles. Also, bringing more diversified skilled individuals helps to broaden the thinking and working perspectives bringing the room to flexibility and idea generation.

2.1.3 Cluster Hire

Hiring group of individuals, to meet requirements of a set of projects within the budget of the organization is called Cluster Hire. For example, if a company wants to group experts for five projects, each of which have predetermined skills required for its completion, the task here would be to find experts that possess these skills. This is called cluster hiring.

The selection process is influenced by various factors such as budget assigned for the set of projects, profit earned on each project's completion, communication cost between the experts in the team, working capacity of each, etc. This concept is getting wide acceptance and importance in the market given that more organizations are working in a project driven environment nowadays. Also, hiring a set of individuals that covers various projects is better than hiring a fixed set of individuals for each project. This approach ultimately helps bring down the Budget factor

Factors to be kept in mind while forming a team

1)Budget

Budget is an upper limit monetary quantity applied by the company towards the hiring cost of individuals for a set of projects. The goal of the organization is to select a team of experts in a way that the budget is not overspent.

2) Experts and their attributes

Expert's Skills: Each expert or individual in the job market or working in a company is known for his expertise in a particular set of skills. We aim at selecting individuals for the projects depending on the skills the experts can bring along with them, that will help in producing a faster and better outcome for the project.

Expert's cost: Each expert demands some monetary benefit in return of the skills he/she provides. We need to keep in mind that the sum of the hiring cost of the experts selected should not be greater that the budget assigned for hiring a team for a set of projects

Expert's capacity: An expert can have more than one skills. In situations that there exist multiple skills that an expert possesses, which are useful for the set of projects we will assign the experts only those skills which are in his working capacity. For example, an expert has six skills that are useful for the given set of projects but the expert has the capacity to work only in three skills. We will make sure that the expert is not overburdened and is assigned skills per his working capabilities.

Experts' collaboration cost: In this special setting, we also make sure that the experts selected have the best collaboration, that is the communication frequency between them. We use their prior collaboration and instances when the two individuals worked together, to calculate the communication cost between them. Forming such a team of experts helps in reducing the overhead time spent in understanding each other's' background etc.

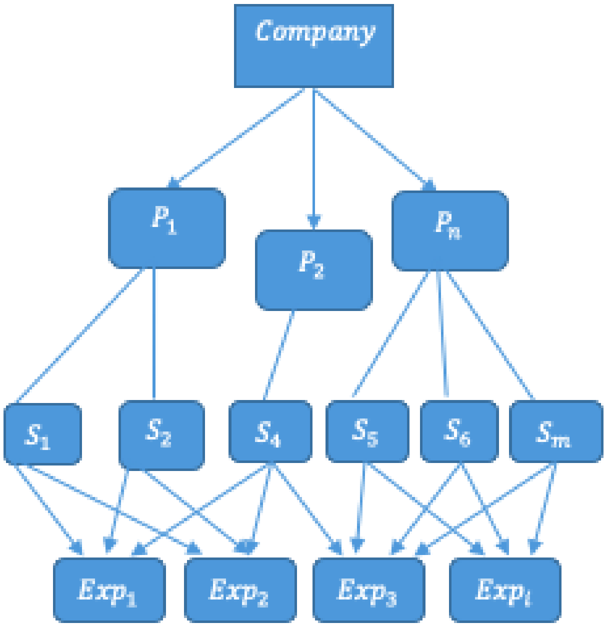

Figure 2 Cluster Hire

3)Project and their attributes

Project's Skills: Each project consists of predefined skills required to produce the desired output. We seek for individuals who have good level of expertise in the skills required by the project.

Skill's demand for experts: We also have to make sure that we find no less than the number of experts demanded to complete a particular skill required for the project. For instance, a project demands two individual who are experts in Artificial Intelligence and three experts in Database Management. Hence, our task is to find that number of expert skill pairs demanded by the project for its successful completion

Project's Revenue: Keeping all the above factors in mind , the group of experts selected for each project helps in producing revenue for the project. Each project has a predetermined revenue (monetary profit) associated to it, which the company obtains after successful completion of the project.

The aim of our approach is to assign enough experts for the set of projects to obtain the maximum profit. While doing so we make sure all the above-mentioned constraints are not violated and the team formed is the most collaborative.

2.2 Fundamental Concepts

2.2.1 Linear Programming

Linear Programming is a mathematical model used for solving decision problems having many decision variables that are limited by a set of constraints. For a linear program the constraints and objective functions are required to be linearly related to the variables of the problem.

Linear Programming Terminology

1.DECISION VARIABLE: These variables determine the quantity that the decision makers would like to determine. Typically, their optimal values are determined with the optimization method. In general, the decision variables are represented using the algebraic notations like

$$x_1 , x_2 , x_3 , x_4 \ldots \ldots x_n$$

2.CONSTRAINTS: A constraint is an inequality or equality defining limitations on decisions. In general, an LP is said to have m linear constraints that can be stated as

$$\sum_{i=1}^{n} a_{ij}\, x_i \leq \; b_i \;\; for\; j = 1 \ldots \ldots m$$

3.OBJECTIVE FUNCTION: The objective function ***minimizes or maximizes*** the quantitative criterion such as cost, profit, utility, or yield. The general linear objective function can be written as $Z = \sum_{i=1}^{n} c_i\, x_i$

4. SIMPLE UPPER BOUND: associated with each value of x_i there may be a specified quantity, u_i that limits its value from above;

$x_i \leq u_i \quad for\ i = 1 \ldots..n$

When a simple upper bound is not specified for a variable, the variable is said to be unbounded from above

5.NONNEGATIVITY RESTRICTIONS: In most practical problems, the variables are required to be nonnegative;

$$x_i \geq 0, for \text{ i=1}\ldots.\ n$$

This special kind of constraint is called a no negativity restriction.

6.COMPLETE LINEAR PROGRAMMING MODEL: Combining the aforementioned components into on single statement gives:

Minimize $Z \quad Z = \sum_{i=1}^{n} c_i\, x_j$

Subject to

$$\sum_{i=1}^{n} a_{ij}\, x_i \leq b_i \ for\ j = 1 \ldots\ldots m$$

$$0 \leq x_i \quad \leq u_i \ for\ i = 1 \ldots n$$

The constraints, including the non-negativity and simple upper bounds, define the feasible region of a problem.

2.2.2 Integer Linear Programming(ILP)

Linear Programming (LP) is a mathematical programming approach where the problem is modeled to find either a minimization or a maximization solution to a function, given certain constraints. Integer Linear Programming (ILP) is a branch of Linear Programming that restricts all the variables of the model to be only integers. A general Integer linear programming model can be represented as

$$Max\ cx$$

$$Ax = b \text{ ……... (2)}$$

$$x \geq 0\ integer$$

All entries of c, A and b are assumed integer. Equation 2 provides one possible formulation of an ILP problem. Alternatively, we have minimization problem or problems with inequality constraints.

Minimization problems can be translated to a maximization problem simply by using the following notation [18]

$$-min\,(-f(x)) \;=\; max(f(x))$$

Inequality constraints can be converted into equality by adding auxiliary variables. For example

$$ax \;<=\; b \; can\; also\; be\; denoted\; as\; ax \;+\; s = b,$$

$$and \quad\; ax >= b \; can\; also\; be\; denoted\; as\; ax - t = b$$

The variables s and t are known as slack or surplus variables.

An important special case of the ILP problem is called binary ILP problem described by

$$Max\; cx$$

$$Ax \;=\; b \; \ldots\ldots\ldots (2)$$

$$x \;\geq 0\; binary$$

Here, x binary means x_i=0 or x_i=1 for all i

2.3 Application of Linear Programming

Integer Linear Programming has found its way as one of the leading methods to find optimal solutions in the field of Operation research. Well known problems like resource allocation, task assignment, resource scheduling can be solved using Integer Linear Programming techniques. Below are some of the classic set of applications where ILP has created a benchmark as a problem solver.

1)Computer Science: One of the largest problems that organizations are facing in today’s world is dealing with large data set and finding ways to gain important insights about the data which can help in better prediction or analysis of the future trend of a company’s sales or profit generation. The process of identifying patterns and meaningful insights from the large dataset is called data mining. Data mining techniques help organizations generate revenue and make concrete decisions on the basis of discoveries and predictions made using the data. Linear Programming provides techniques to identify such pattern and make data more sensible and useful for prediction and analysis. The process of Integer Linear programming involves formulating an ILP model for the problem, using decision variables and constraints which are supplied to the objective function.

2)Manufacturing: ILP is one of the most widely used mathematical approach used in the manufacturing industry for resource allocation and profit maximization. Resource planning is important for the manufacturing industry which requires proper planning of the optimal mix of products that are needed to be manufactured. One of the most common ILP techniques that help in resource planning and manufacturing is a simplex method which uses the concept of slack variables, tableaus and pivot variables for finding the optimal solution to an optimization problem

3)Transportation optimization: The transportation systems depend on linear programming for getting solutions that give cost and time efficient results. Almost all transportation systems treat scheduling of shipments, passengers and in-time delivery as their key priorities. Airlines apply linear programming techniques by setting the cost and travel demand as factors to maximize their profits. In a same way, bus transport officials use LP for scheduling routes and allocating drivers. Optimization using ILP surges efficient decision making and reduces expenses.

4)Energy Industry: In today's modern grid management system, we not only involve the traditional electric generation source but also those from the non-renewable sources such as wind and photovoltaics. Thus, it becomes important to optimize the load between requirements, transmission, generators and distribution lines. Linear Programming has been used in the electrical industry as a method to redesign the power system design that helps to match the electric load given the demand for electricity and its generation time.

2.3.1 Example of Integer Linear Programming

Capital Budgeting Problem

A firm has n projects to undertake but, because of budget restrictions, not all can be selected.

Project j has a present value of c_j, and required an investment of a_{ij} in the time period i, where $i = 1 \dots \dots, m$. The capital available in time period i is b_i. The problem of maximizing the total present value subject to the budget constraints can be written as [11]

$$\text{Maximize} \quad \sum_{j=1}^{n} c_j\, x_j$$

Subject to

$$\sum_{j=1}^{m} a_{ij}\, x_j \le\ b_i \ \ for\ i = 1 \dots \dots m$$

$$0\ \le\ x_j \quad \le u_j\ \ for\ j = 1 \dots n$$

where x_j=1 if the project j is selected and x_j=0 if the project j is not selected

The fixed charge problem

In general, the cost of an activity is a special nonlinear function of the activity level x given by

$$f(x)=\begin{cases} d + cx\ if\ x > 0 \\ 0\ \ if\ x < 0 \end{cases}$$

If $d > 0$ and f is to be minimized, we have the problem

$$\begin{array}{ll} & \min cx + dy \\ & x \ge 0 \\ \text{subject to} & x - vy \le 0 \\ & y = 0{,}1 \end{array}$$

where *y* is an indicator of whether or not the activity is undertaken, and *u* is a known, finite, upper bound for *x*. The second constrain guarantees that $x > 0$ implies $y=1$

The plant location problem

Consider n customers, the $j-th$ one requiring b_j units of a commodity. There are m locations in which plants may operate to satisfy the demands.

There is a fixed charge of d_i for opening plant i, and the unity cost of supplying customer j from plant is c_{ij}. The capacity of plant i is h_i.

The ILP model formulated is

$$\min \sum_{i=1}^{m} \left(\sum_{j=1}^{n} c_{ij} x_{ij} + d_i y_i \right)$$

$$\sum_{j=1}^{m} x_{ij} = b_i$$

$$\sum_{j=1}^{n} x_{ij} - h_i y_i \leq 0$$

$$x_{ij} \geq 0, \qquad y_i = 0,\ 1.$$

The knapsack problem

Suppose n different types of scientific equipment are taken into consideration for the inclusion on a space vehicle. Let c_j be the scientific value per unit of a_j the weight per unit of the j-th type. If the total weight limitation is b, the problem of maximizing the total value of the equipment taken is

Maximize $Z \quad \sum_{j=1}^{m} c_j \, x_j$

Subject to

$$\sum_{j=1}^{n} a_j \, x_j \leq \ b$$

$$x_j \geq 0, integer$$

where x_j is the number of units $j-th$ type included.

2.4 Algorithm

The Algorithms used for Integer Linear Programming Problems rely on two basic concepts:

1. LP Relaxations

Removing the integrality constraints from the Integer Linear Program is called relaxation. This allows the variables of the ILP model to take on non-integral values. For example, $x_i \epsilon \{0,1\}$ can be replaced by two continuous variable, for $x_i \geq 0$ and $x_i \leq 0$ respectively. Hence the resulting relaxation for the Integer Linear Programing problem is a Linear Program. This technique will help transform the NP-Hard optimization problem to a related problem which is solvable in polynomial time providing near- optimal solution to the original problem. LP relaxation can be used to gain information about the solution to the original integer program.

2. Bounds

LP relaxation technique solves the ILP problem and provides LP solution which gives an upper bound to the integer linear problem. The ILP problem is solved to find the best solution in the feasible region that produces a result near the upper bound obtained by LP relaxations.

There are three main types of Algorithms used for Integer Linear Programming

a) Exact Algorithm:

Classical approaches to exactly solve a combinatorial optimization problem are called exact algorithm. They do guarantee to find an optimal solution but may take an exponential number of iterations. Some of the examples of exact algorithms popularly known are cutting planes, branch and bound and dynamic programming

b) Heuristic Algorithms

Heuristic Algorithms are those algorithms that provide suboptimal solution but do not guarantee the quality of the solution obtained. Although the computation time is not guaranteed to be polynomial, it is known from empirical evidences that some of these algorithms find good solutions.

c) Approximation Algorithms

These are the algorithms that provide suboptimal solution in polynomial time together with a bound on the degree of sub-optimality.

2.4.1 Exact Algorithm

2.4.1.1 Branch and Bound Algorithm

Branch and Bound Algorithm uses a ‘divide and conquer’ approach to explore the set of feasible solutions. Its uses the concept of *bounds on the* optimal cost *in* order to avoid exploration on the entire feasible integer solutions. The algorithm consists of systematic enumeration of multiple solutions, which is similar to a rooted tree. Each branch of this tree is explored, which represents subsets of the solution set. Before selecting any branch, it checks the value with the upper and lower bounds on the optimal solution, and is discarded if it cannot produce a better solution than the best one found so far by the algorithm.

The steps that the algorithm uses to determine the optimal integer solution for an ILP model can be listed as follows [8]

1. Find the optimal solution to the linear programming model with the integer restrictions relaxed.
2. At node 1, let the relaxed solution be the upper bound and the rounded-down integer solution be the lower bound.
3. Select the variable with the greatest fractional part for branching. Create two new constraints for this variable reflecting the partitioned integer values. The result will be a new ≤ constraint and a new ≥ constraint.
4. Create two new nodes, one for the ≤ constraint and one for the ≥ constraint.
5. Solve the relaxed linear programming model with the new constraint added at each of these nodes.
6. The relaxed solution is the upper bound at each node, and the existing maximum integer solution (at any node) is the lower bound.

7. If the process produces a feasible integer solution with the greatest upper bound value of any ending node, the optimal integer solution has been reached. If a feasible integer solution does not emerge, branch from the node with the greatest upper bound.
8. Return to step 3.

Some of the disadvantages of the Branch and Bound algorithm are

1. Branching into sub problems and then solving them resulted in heavy computational time
2. Branching on the node with the smallest bound increased the storage space
3. Running time grows exponentially with the size of the problem but small moderate problems can be solved in reasonable time

To overcome the abovementioned limitations and disadvantages Branch and Cut algorithm was introduced which is explained in detail below.

2.4.1.2 Branch and Cut Algorithm

Branch and Cut method [15] is one of the most powerful and successful algorithm for solving integer linear programming problems. It is an exact algorithm consisting of a cutting plane method and a branch and bound algorithm. The branch and cut algorithm uses two techniques

1.Cutting Planes: To solve an ILP, first the relaxation of the problem is considered to cut away parts of the polytope by adding new constraints to bring the best integer solution. In order to cut the planes, the valid inequality of the ILP which causes violation is studied and then separated.

2.Branch and bound: All variables are arranged similar to an enumeration tree, then this tree is partially traversed to compute global lower bounds and local upper bounds, which are used to avoid parts of the trees that do not produce the optimal solution.

The below figure shows the general strategy used by Branch and Cut algorithm [13]

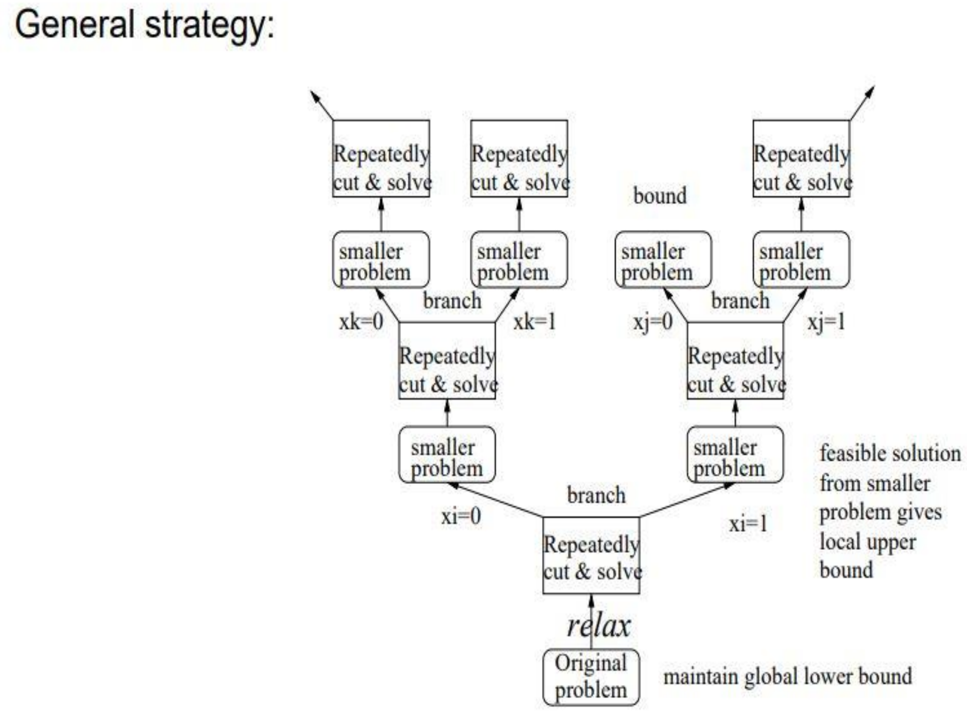

Figure 3 ILP using Branch and Cut solver

2.4.2 Heuristic Algorithms

The term heuristic is used for algorithms that find the best solution among all possible solutions, but they do not assure that the best solution is found. The approach tries to find near to optimal or an optimal solution in some case. Depending on the heuristic logic used, the algorithm decides which branch should be followed in order to reach the best possible solution for the given problem. The decision is made depending upon the available information in each step.

The only drawback of this approach is that it compromises with the optimality, accuracy of the result returned by the algorithm and completeness. The advantage of using this approach is that it provided quick results, although the optimality of the result obtained by the algorithm cannot be guaranteed.

A well-known example of a heuristic algorithm is used to solve the Travelling Salesman Problem. The problem is as follows: given a list of cities and the distances among the cities, which is the shortest route possible that can help visit as many cities as possible exactly once. A heuristic algorithm used to quickly solve the problem is by using Nearest Neighbor algorithm(NN) in which we start by randomly picking a city and then finding the closest city. The remaining cities are then visited until the closest city is found. Following are the steps of the NN algorithm.

1.Start at a random vertex from the given set of vertices.

2.Determine the shortest distance that connected the current vertex and an unvisited vertex V

3.Make the current vertex, the unvisited vertex

4.Make V as the visited vertex

5.Note the distance travelled

6.Terminate if no other unvisited vertices remain

7.Repeat step 2 to 5

This algorithm is heuristic such that it does not take into account the possibility of better steps being excluded due to the selection process.

2.5 Literature Review

2.5.1 Team formation Problem and cost functions for best team formation in Social Networks

The author Lappas et. al [1] were the first to introduce team formation problem in the community of data mining and data management. They gave two functions for estimating the value of the communication cost of a team. The first function determines the largest shortest path between any two nodes of the subgraph formed. The second function takes into account the cost of the minimum spanning tree (MST) on the subgraph. These functions had certain drawbacks and were then improved by Kargar et. al [2] where they introduced two new cost functions. The first function known as the *sumofdistance* function discovers the team communication cost based upon the summation of the shortest distance between the experts who possess the skills required for the project. The second function involves the communication cost of the leader with all other experts of the team and is known as the *leaderdistance* function which computes the sum of shortest path between the leader and each expert in the project. We use the above *sumofdistance* function and translate the team formation problem into a mathematical model using Integer Linear Programming approach.

2.5.2 Social Network an important factor for efficient team formation

Effective team organization is strongly related to the structure that is considered for team formation to accomplish complex tasks. The author Matthew et. al. [3] prove how such social structures play a vital role in task completions and successful completion of project work. Each project and team is modeled by a subgraph where the set of vertices represent the expert that has the collective skills required for the fulfillment of the tasks involved. They then run experiments to study the network effects on the real-world data and shows how social network amongst the team members affect the team performance

The dynamics of group formation is studied by authors Lars et. al. [4], where they consider three parameters: membership, growth, and evolution, for analyzing the evolvement of large groups in an organization/community. They study how various factors influence the desire of an individual to join a community and how companies perform research on these factors for their scalability. It is said in this paper that the willingness of a person to join a community depends upon the number of individuals he is friends with, in that organization. Decision-tree techniques are used to identify the most prominent determinants influencing such social group. They further device a novel methodology that measures the incoming and outgoing of individuals from one community to another depending on their level of interest in a particular topic. The experiment and results are obtained by running the proposed method on DBLP dataset.

2.5.3 Genetic Algorithm for team formation

Integer Linear Programming is used to model the team formation model along with constraint based on knowledge and collaboration as factors that affect the objective function. The familiarity score is calculated on the average of the time two persons have been knowing each other. In the context of DBLP data set, the authors used the previously published papers and the time interval in between to formulate the familiarity score. Similarly, the knowledge score is obtained on the basis of the keywords extracted from the papers published by these authors. The authors Chien et. al. [5] focus on assigning a good team manager on the basis of analyzing the knowledge he/she has and selecting other team members such that there exist a good synergy required for successful and timely completion of the project. Since it is an NP-hard non-linear problem, a genetic algorithm is proposed for problem-solving

2.5.4 Team formation problem solved using various techniques

The author Adil et. al. [7] in this paper proposes an optimization model consisting of objective functions limited to constraints based on the criteria that drive the best team selection for a project. They study various human and nonhuman factors that affect the health of a project and formulate the subjects that drive the decisions for the formation of a team. The mathematical model aims at maximizing the compatibility among the team members based on factors such as the cost of the expert, team size, budget allocated for the team, number of activities involved in the project etc. The fuzzy objectives and constraints are then translated to the simulated annealing algorithm which produces the desired output.

In [16], the author Xinyu et. al. solves the team formation problem considering various approaches. Each approach uses different applications for forming a team of experts.

1. R-TF Algorithms: This approach was used to hire a team of well-known experts for the purpose of reviewing a paper.
2. Steiner-TF algorithm: This algorithm suggested that the team of reviewers hired can give diversified results, as the reviewers come from a wide range of background.
3. SD-TF algorithms: The author's suggested that this approach can be used to hire experts seeking a rise in their positions and are new to the industry.
4. LD-TF algorithms: It makes sure that while assigning experts to skills, the working capacity of the experts is not overloaded

All the above approaches use a single attribute or feature for forming of team which makes it slightly unrealistic, since for each definition we will need to use different approach to get a solution.

2.5.5 Clustering Technique for Team Formation Problem

Team Formation Problem(TFP) in social network has been studied widely by many authors and researchers. Many techniques were proposed to solve TFP, for discovering experts that could cover all the tasks required for a project and that the team formed is the most collaborative. With this objective Kalyani et al. [17] in their paper propose clustering technique for grouping set of experts in form of a cluster such that the cluster points consist of experts that hold the skills required for the project. They've adopted weighted SCAN algorithm, which is an enhanced version of SCAN i: e Structural Clustering Algorithm to solve TFP with minimum communication cost. The communication cost is the cost associated between two experts on the basis of their past collaboration or prior work experience in one or more projects. The basic idea of their technique is to identify clusters, hubs and outliers in the network of experts where every node represents the expert and the edge weights define the communication cost. They first approach the problem by finding the pool of experts that can cover the skills required for the project. Then, they search for the highly connected expert among all experts and declare it as the core. The cluster is then expanded from the core to the neighboring nodes i: e from densely connected nodes to loosely connected nodes. The cluster is expanded till the threshold range of communication cost is reached. Comparison of weighted scan is done with other greedy, genetic, random and exact algorithm and present the analysis of the results achieved.

2.5.6 Online Team formation in Social Networks

Anagnostopoulos et al. [10] in their paper study the online version of the team formation problem. They use the social network of experts which possesses set of skills and have compatibility with each other obtained from prior work experience. They consider a sequence of tasks that arrive in an online fashion, each task requiring specific skills. The idea behind this paper is to form multiple teams for multiple projects, each project and the set of tasks for the project appearing in an online fashion. They aim at forming the team keeping in mind three attributes. Firstly, the skills required for project completion are covered. Secondly. the team formed is collaborative and thirdly, the working capacity of the individual in the team is not overloaded. Various heuristic and approximate algorithms are used to solve the TFP problem with the above objectives followed by result analysis and conclusions.

2.5.7 Cluster Hiring Problem

The first to study the Cluster hire problem were Golshan et al. [13] for Hiring a group of experts, each possessing a one or more skills, for a set of projects where each project consists of a set of tasks that requires specified skills. This is referred to as the CLUSTER HIRE problem. They study the variants of this problem by making realistic assumptions that affect the cluster hiring process. The aim of the paper is to hire a group of individuals for a set of projects, where the team formed maximizes the total profit and is within the budget allocated for hiring the group of experts for project completion. Further to this, each individual's working capacity is kept in mind and it is made sure that the work load is not exceeded for the individual working in multiple projects. They propose two variants of greedy algorithm and use real data sets to perform experiments and result analysis. While Golshan et al. study the problem of cluster hire with an objective to maximize the profit, Meet et al. [14] were the first to study CLUSTER HIRE problem in social network. The aim of the paper is bi-objective, where they propose to hire a group of individuals which are the most collaborative and give the maximum profit on successful completion of the project. They assume that there exists a social network among experts where each expert is associated to another through the prior work experience in one or more common projects. Each expert in this setting is associated by salary, i:e., the hiring cost, and the working capacity. The hiring is made keeping in mind that the sum of the salary of the experts selected doesn't exceed the budget allocated for the set of projects and that no experts in the team is overloaded. To achieve all the above attributes of the team selection, the bi-objective problem is then solved by proposing two greedy algorithms, that is Expert Pick Strategy and Project Pick strategy. Both of these algorithms use scoring function. The Expert pick strategy scores each expert in the network that can cover one or more skills required by the projects such that it covers many skills to get the most profitable project, the hiring cost is cheap and the communication cost of the expert is least. Using this approach, the experts with the highest score is then chosen. Since it is less possible that a cheap expert that covers many skills can have the most collaboration, they introduce λ a trade-off parameter between profit and communication cost which has a value between 0 and 1. This trade-off helps in determining whether to put more weight towards profit or communication cost. The second strategy called the Project pick strategy uses the same strategy to score projects that have high profit, the set of experts required to complete the project are cheap and the set of experts selected are able to communicate well. They further run experiments on real dataset and present the efficiency of their approach when compared with the random algorithm.

2.6 Problem Statement

Let $E = \{e_1, e_2, e_3, \ldots\ldots e_n\}$ denote a set of n experts and S =$\{s_1, s_2, s_3, \ldots\ldots s_m\}$ denote the m skills. All symbols used in this thesis are summarized in Table 1. Each expert i possesses a set of skills and we use CS_{iu} is whether expert i possesses skill u by CS_{iu}. Each expert i holds a monetary cost(salary) to perform task associated to the project he is assigned. This is measured by dollar value denoted by C_i. In this setting we also use a set of projects $P = \{p, p_2, p_3, \ldots\ldots p_k\}$. Each project also consists of required skills and the number of experts needed per skill in order to successfully complete the project. This set is represented using $PSRup$ for a project p . Also, each expert i can offer his/her expertise at most Cap_i times (i:e., capacity of expert i). This is because our aims are to assign experts skill in such a way that they don't get overloaded when assigned to multiple projects. Hence, each expert has a maximum capacity to participate. We add this as a constraint to our ILP model.

Definition 1: Group of Experts

Given a set of n experts E, a set of m skills S, and set of k projects P, a group of experts' E is able to complete a subset of projects P, if the following holds.

1. An expert *e* is assigned to perform the required skills *s* for *p*.
2. The number of experts assigned per skill should be the same as the demand *r* of the skill *s* for each project *p*, where the same expert *e* cannot perform the same skill *s* in the same project p twice
3. Each expert *e*, should not be assigned more than Cap_i skills
4. The sum of C_i among those selected should be less than budget *B*
5. The total communication cost is the lowest, meaning they have the highest collaboration.

The experts are connected with each other in a network which is modelled in an undirected graph G. Each experts i is denoted as a node of graph G. Two experts in a graph are connected to each other via edge weight with the weight denoting the communication cost between these two individuals. The communication cost is extracted from the prior collaboration in the past (i.e. participation in a same project). In this setting, the edge weight denotes the strength of collaboration between the experts. The smaller the communication cost i.e. the edge weight, the stronger the prior collaboration between the experts. For an instance if two experts have participated in four prior projects, their communication cost would be less than that of the experts who have collaborated for prior two projects. When no collaboration is presented between experts the weight of the shortest path between them in G is taken to determine the communication cost between them. Similar to the previous understanding for most collaboration, the shortest the distance between the individuals the maximum the two experts can collaborate. Here, we aim to choose a group of experts which have minimized communication cost among them.

Example of a social network graph of Experts

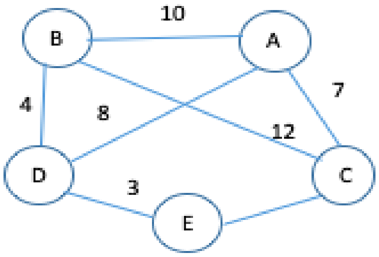

Figure 4 Social Network of Experts

Definition 2: Communication cost

The communication cost between two experts, i and j is calculated in a similar way as given by Kargar et. al [2] known as the Sum of Distance function represented as below,

$$CC(E) = \sum_{i=1}^{|E|} \sum_{j=i+1}^{|E|} Dist(e_i, e_j)$$

Completing each project brings a profit in dollar value which is shown by $Revenue(p)$. In our research work the goal is to select the set of projects that maximize our profits which is obtained by the summation of the revenue of the selected/covered projects.

Definition 3: Profit of the projects:

Given a set of projects P, the profit received when completing these projects is defined as follows:

$$Profit(P) \ = \sum_{p=1}^{|P|} Revenue(p)$$

where revenue function $Revenue(p)$ maps each project with an integer number denoting a dollar amount obtained by completing the project.

For performing the set of projects, we are also provided a predefined budget B. While selecting the set of projects that provide maximum profit, we also make sure that the total hiring cost which is the salary of each expert is within the budget allocated for the set of projects.

In our study, we provide two ILP models, one that returns group of experts with maximized profit and another selecting a subset of projects that return maximum profit and have minimum communication cost amongst the experts selected to perform the project

Problem 1: Given set of n experts E, set of m skills S, a set of k project P we are interested in choosing a group of experts E’ and a subset of projects in P’ so that the following objective is maximized:

$$Profit(P') \ = \ \sum_{p=1}^{|P'|} Revenue(p)$$

Problem 2: Similar to problem statement 1, we propose to solve the cluster hiring in social network where we choose a group of experts E’ and a subset of projects in P’ in P, such that the following objective is maximized:

$$Profit(E', P') = \ \sum_{p=1}^{|P\prime|} Revenue(p) - \sum_{i=1}^{|\mathrm{E}\prime|} \sum_{j=i+1}^{|\mathrm{E}\prime|} Dist(e_i, e_j)$$

The above objective aims at two things: (i) which are, to cover a subset of projects which gives maximized profit and (ii) the group of experts selected to preform over the projects are collaborative and has minimum communication cost amongst them. Here the distance is calculated between experts of the same project and not across the projects

We explain in detail about the ILP models and heuristic algorithm later in Chapter 3

Theorem 1 *Problem 1 is NP-hard*

Proof 1 *Finding a group of experts to cover a subset of projects* P *and maximizing Profit*(P) *under the given budget B is proved to an NP-hard problem in kargar et al.[2] Since the objective of the present problem 1 is linearly related to* $Profit(p)$*, the present optimization problem 1 is also NP-hard.*

CHAPTER 3
IMPLEMENTATION DETAILS

In this chapter, we propose two different strategies for finding a group of experts which produces maximized profit and has the most collaborative team of experts, i.e. having minimized communication cost. The first strategy is using a heuristic approach which is greedy and covers the highest profitable project in each iteration. While in the second approach we use ILP solvers which gives an exact solution to the given problem and finds the best feasible group of experts from the available input graph of nodes. We further create two ILP models, one which produces maximum profit i.e. covers projects with highest profit, and the other which covers profitable projects as well as selects experts that have the highest collaboration with other experts in the project.

As previously stated, our main objective behind this study is to find a group of experts which covers maximum projects under given budget such that the group of experts selected has the maximum collaboration when deployed to work for the selected subset of projects.

In our study, we suggest two models

a) The model aims at selecting the group of experts with the objective of covering projects which produce maximum profit.
b) In this scale, the above model also picks experts based upon their collaboration with other project members at the same time covering project with high profit returns.

3.1 Notations used in ILP

In this section, we outline the notations used to formulate the problem with the objective discussed in the former section of this chapter.

E	Set of n experts $\{\mathrm{e}_1, \mathrm{e}_2, \mathrm{e}_3, \ldots ..\, \mathrm{e}_\mathrm{n}\}$
S	Set of m Skills $\{\mathrm{s}_1, \mathrm{s}_2, \mathrm{s}_3, \ldots ..\, \mathrm{s}_\mathrm{m}\}$
G	Input graph G that models the social network
P	Set of k Projects $\{\mathrm{p}_1, \mathrm{p}_2, \mathrm{p}_3, \ldots ..\, \mathrm{p}_\mathrm{k}\}$
$\mathrm{Dist}_{\mathrm{ij}}$	Communication cost between expert i and expert j
$\mathrm{CS}_{\mathrm{iu}}$	whether expert i posseses skill u
$\mathrm{Cap}_{\mathrm{i}}$	Capacity of expert i to offer her expertise
$\mathrm{PSR}_{\mathrm{up}}$	Number of experts required for skill u in project p
W_p	a binary variable that represents whether project p, has been selected or not
C_i	Hiring cost of each expert i
X_i	a binary variable which represents if an expert i, has been selected or not
Budget	Total budget of hiring a group of experts
$\mathrm{Revenue}_\mathrm{p}$	Profit of each project p
$\mathrm{V}_{\mathrm{iup}}$	*Binary* Decision variable fo whether expert i and its skill u has been selected for a project p
$\mathrm{Z}_{\mathrm{iujvp}}$	Binary Decision variable for selecting the two experts i and j having skills u and v that covers for a project p
$\mathrm{Y}_{\mathrm{iujvp}}$	Binary Decision variable for selecting the two experts i and j having skills u and v that covers for a selected project p

Table 1 Notations in ILP

3.2 ILP model for maximized profit

We first present the ILP model which returns the group of experts covering a subset of projects which produce maximum profit and do not guarantee the experts to have good collaboration amongst each other. In the description below, we show the objective function and the set of constraints. The constraint number with sub notations are used to describe the expert skill and project combination.

Objective Function:

$$maximize\ \sum_{p=1}^{k}(\mathrm{W_p} \times \mathrm{Revenue_p}) \qquad (3.1)$$

Subjected to:

1. The value of the binary variable $\mathrm{V_{iup}}$ is 1 when it matches the value set in the input $\mathrm{CS_{iu}}$, otherwise is set to value 0

$$\forall \mathrm{u} \in \mathrm{S}, \forall \mathrm{i} \in \mathrm{E}\ \ \sum_{p=1}^{k} \mathrm{V_{iup}} \leq \ \ \mathrm{CS_{iu}} \qquad (3.2)$$

2. For all the projects covered, the number of experts selected should not be less than the number of experts demanded by each skill of that project

$$\forall \mathrm{p}\ \in \mathrm{P}, \forall \mathrm{u}\ \in \mathrm{S}\quad \sum_{i=1}^{n} \mathrm{V_{iup}} \geq \mathrm{PSR_{up}} \times \mathrm{W_p} \qquad (3.3)$$

3. The expert selected should not be assigned a number of skills more than its working capacity. The capacity is provided as an input to the ILP model

$$\forall i\ \in E\quad \sum_{u=1}^{m}\sum_{p=1}^{k} \mathrm{V_{iup}}\ \leq \mathrm{Cap_i} \qquad (3.4)$$

4. Expert i is assigned to some project for some skill whenever there exists a project p and a skill u that he/she is assigned to,

$$\forall \mathrm{p}\ \in \mathrm{P}, \forall \mathrm{u}\ \in \mathrm{S}\,, \forall \mathrm{i} \in \mathrm{E},\quad \mathrm{X_i} \geq \mathrm{V_{iup}} \qquad (3.5)$$

5. The sum of value of $\mathrm{V_{iup}}$ for every project and skills should be greater than or equal to the value of $\mathrm{X_i}$

$$\forall i\ \in E \qquad \mathrm{X_i} \leq \sum_{u=1}^{m}\sum_{p=1}^{k} \mathrm{V_{iup}} \qquad (3.6)$$

6.The sum of the hiring cost of the experts should be less than the given budget

$$\sum_{i=1}^{n} \mathrm{C_i} \times \mathrm{X_i} \leq \mathrm{Budget} \qquad (3.7)$$

3.2.1 Justification of the ILP

As explained earlier, the aim of our algorithm is to cover highest profitable projects under a given budget constraint. The objective function is thus to maximize the profit calculated by the summing up the revenue of each selected project $W_p$whose value is 1. Constraint 3.2, corresponds to the value of the binary variable V_{iup} being 1, if an expert *i* is assigned to skill *u* in project *k* only if the expert possesses skill u. In constraint 3.3, we make sure that for every project and the skill associated to that project, we hire exactly the same number of experts as demanded per skill listed in the input PSR_{up} . This also sets the value of variable W_p as 1 shows whether the project has been covered and assigns the experts holding the skills and demand of the number of experts required. Constraint 3.4 is provided to make sure that the sum of the V_{iup} variable for each project and skills for the project should be less than or equal to the working capacity $Cap_i$for expert *i,* which is given as an input to our model. In our setting, we also want to make sure that the hiring cost of an expert is deducted only once from the budget no matter how many skills and in how many projects the same expert is assigned. To make sure this happens, we include constraints 3.5 and 3.6. Constraint 3.5 makes sure that the sum of experts in X_i should be greater than or equal to the value of V_{iup} for each project, skill and expert. Basically, we make sure that X_i value is set to 0 for every expert in Viup whose value is 0. On the other hand, X_i value should be less than or equal to the sum of every projects and skill. This is to assure that the value of X_i is set to 1,if expert *i* is assigned to any project for any skill. In 3.7, we sum the salary of all the selected experts to make sure it is less than or equal to the given budget which is given as an input to the model.

All of the above explained constraints ensure the below attributes to solve the cluster hiring are taken into consideration

1)A same expert is not assigned more than once to the same skill in the same project

2) Hiring cost of the expert is subtracted only once from the given budget given that an expert can perform up to *n* number of skills in *m* number of projects

3)No expert is assigned to skills more that its allocated capacity

3.3 ILP model with maximum profit and minimum communication cost

The ILP given below is proposed for solving the Cluster Hire problem in the social network. We choose to work using ILP technique, since it goes through all the possible solution nodes to return the best solution for the given objective function. Here, we also prove the scalability of the existing model by showing how the ILP model for cluster hiring with maximized profit can be modified to include the communication cost feature ensuring the group of experts selected return maximized profit and have good communication amongst each member in the project he/she is assigned to

Objective function:

$$\sum_{p=1}^{k}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{u=1}^{m}\sum_{u\neq v}^{m} \quad (W_p \times \text{Profit}_p) \quad -\left(Dist_{ij} \times Y_{iujvp}\right) \qquad (3.8)$$

Subjected to:

1. The value of the binary variable V_{iup} is 1 only when expert i possesses skill u

$$\forall u \in S, \forall i \in E \;\; \sum_{p=1}^{k} V_{iup} \leq \quad CS_{iu} \qquad (3.9)$$

2. For any project covered, the number of experts selected for this project should be equal to the number of experts demanded by each skill of that project

$$\forall p \;\in P, \forall u \;\in S \quad \sum_{i=1}^{n} V_{iup} \geq PSR_{up} \times W_p \qquad (3.10)$$

3. The expert selected should not be assigned skills more than its working capacity. The capacity is provided as an input to the ILP model

$$\forall i \;\in E \quad \sum_{u=1}^{m}\sum_{p=1}^{k} V_{iup} \;\leq Cap_i \qquad (3.11)$$

4. The value of V_{iup} should be less than or equal to the value of X_i

$$\forall p \;\in P, \forall u \;\in S\,, \forall i \in E, \quad X_i \geq V_{iup} \qquad (3.12)$$

5. The sum of the value of V_{iup} for every project and skills should be greater than or equal to the value of X_i

$$\forall i \ \in E \qquad \mathrm{X_i} \leq \sum_{\mathrm{u=1}}^{\mathrm{m}} \sum_{\mathrm{p=1}}^{\mathrm{k}} \mathrm{V_{iup}} \tag{3.13}$$

6. The sum of the hiring cost of the experts should be less than the given budget

$$\sum_{\mathrm{i=1}}^{\mathrm{n}} \mathrm{C_i} \times \mathrm{X_i} \leq \mathrm{Budget} \tag{3.14}$$

7. To calculate the communication cost between two experts we write the below constraints. Since we need to multiply the two variables $\mathrm{V_{iup}}$ and $\mathrm{V_{jup}}$ which would make the model non-linear we linearize it using the general technique, by introducing the third variable $\mathrm{Z_{iujvp}}$ as explained below:

$$\forall \mathrm{p} \ \in \mathrm{P}, \forall \mathrm{u} \ \in \mathrm{S}\,, \forall \mathrm{i} \in \mathrm{E}$$

$$\mathrm{Z_{iujvp}} \ \leq \ \mathrm{V_{iup}} \tag{3.15}$$

$$\mathrm{Z_{iujvp}} \ \leq \ \mathrm{V_{jup}} \tag{3.16}$$

$$\mathrm{Z_{iujvp}} \geq \ \mathrm{V_{iup}} + \mathrm{V_{jup}} - 1 \tag{3.17}$$

8. We introduce $\mathrm{Y_{iujvp}}$ variable to make sure communication cost is calculated only among the experts within the project selected.

$$\forall \mathrm{p} \ \in \mathrm{P}, \forall \mathrm{u} \ \in \mathrm{S}\,, \forall \mathrm{i} \in \mathrm{E}$$

$$\mathrm{Y_{iujvp}} \ \leq \ \mathrm{Z_{iujvp}} \tag{3.18}$$

$$\mathrm{Y_{iujvp}} \ \leq \ \mathrm{W_p} \tag{3.19}$$

$$\mathrm{Y_{iujvp}} \geq \ \mathrm{Z_{iujvp}} + \mathrm{W_p} - 1 \tag{3.20}$$

For this model, we modify the objective function of the previous ILP model such that along with profitable projects, the solver selects experts having the lease communication cost. Since this is a maximization problem, we use the standard method to turn a minimization problem to maximization by introducing the inverse of it. Hence, the objective function included the profit augmented by the negative of communication cost between experts selected for that project. The constraints from 3.9 to 3.14 resemble those used in previous

ILP model. The new set of constraints added from 3.15 to 3.20 are mainly to calculate the distance/communication cost between the experts. Constraint 3.15 to 3.16 is the linearize form for V_{iup}* V_{jup}. Since this equation would have made the model non-linear, we used the standard way of decomposing multiplication of variables to the above set of equations. Thus Z_{iujvp}value is set to 1 when there exists a communication cost between experts i and j with skills u and v respectively. Also, further here we calculate the communication cost amongst members of each selected project. Again, the original equation $Z_{iujvp} * Wp_p$ was decomposed to linear equations using standard technique introducing the variable Y_{iujvp}

This model for ILP takes care of an additional feature, the which is communication cost, which helps make better hiring decisions for cluster hiring in social network. It ensures all the previously mentioned attributes of cluster hiring with one more attribute to select the individuals which have prior collaboration with other experts in the project and hence performs project tasks with collaboration and co-operation with increased synergy.

3.4 Heuristic Algorithm for Cluster Hire Problem

In this section, we introduce our heuristic algorithm for solving the Cluster Hiring problem with the demanded project skill's demand, i.e. specific number of experts needed per skill per project along with the intent to hire a group of experts that are collaborative and produce maximum profit. The algorithm is designed so to cover as many profitable projects with in the given budget. We assign the score to each uncovered project in each iteration to choose the project which gets the highest score with the hope that our algorithm returns the highest profit.

The score for finding the highest profitable project is designed keeping in mind these intuitions:

1. We want to choose the set of experts that are less expensive, such that they cover all the skills required by the project and the budget is not overspent on a single project.

2. The set of experts selected for the project should be able to communicate effectively with each other

3. Choose the project which returns the highest profit.

We structure the scoring function that takes into account a combination of all the above intuitions. Hence at each iteration, for each uncovered project p, we find a set of experts to cover the required skills for p. Our goal is to select those subsets of experts that cover all the skills and experts demanded per skill per project and also, that no expert covers the same skills more than once for the same project demanding more than one individual to perform the same skill. The algorithm thus aims to find the subset of experts that have minimized hiring and communication cost. In each iteration, we find the set of experts that can cover the project. At the start, we have an empty set; hence, for the skill which appears in the first iteration of the project, we find and assign the expert which covers the skill in the least expense. After the first expert is selected, we then select the experts who have both minimum cost and minimum communication cost with each member in the team selected so far in the project. We then use our scoring function as mentioned below to rank the projects to select the one with the highest profit return

$$Profit(p) = Revenue(p) - min\{C_e + ComCost(e, e') \}$$

Here e, e' represent neighbouring experts in the same project. Our equation thus aims to choose the subset of projects and the team of experts, that produces maximum profit.

The heuristic algorithm is our solution to the cluster hiring problem where we give certain inputs to our algorithm like the graph *G* with nodes representing experts' along with project attributes like profit per project, skills required by each project, number of individuals required per skill per project.

3.4.1 Cluster Hire Algorithm for maximized profit and collaborative team

Input: set of n experts $E = \{e_1, e_2, e_3 \ldots. e_n\}$, set of s skills $S = \{s_1, s_2, s_3 \ldots. s_m\}$ and set of k projects $P = \{p_1, p_2, p_3 \ldots. p_k\}$, graph G that models the network of experts, capacity of experts $Capi$, cost of experts Ci. Project's skills and number of experts demanded to work on that skill $PSR(up)$

Output: Team of experts for subset of projects along with their skills, maximized profit under given budget B

In the first line, we initialize the variables required such as setting the min_cost, budget, temp_team, highest_profit which are responsible for storing the temporary team of experts, the highest profitable project from all the available projects and left-over budget. The while loop in line 2 iterates until the given budget is exhausted and there are projects left to be assigned a team. Line 3,4,5 is to iterate through each project, their skills and number of experts needed per skill for a given project. Thus, we aim at assigning an expert per skills and repeat the process unless the required number of experts are selected for the skill. Line 6 iterates through all the experts and checks if the expert in the current loop has the capacity and holds the skill required by the project in line 7. If the expert holds the skill required by the project, the hiring cost of that expert is stored in the exp_cost variable in line 8. We check to see if there are any expert selected in the temp_team for this project. If not, we find and assign the expert with the least salary offering the skills, that appears the first in the project in line 9. Line 10 to 12, is executed when there exists at least one expert in the temp_team. The collaboration between the expert in the current loop and each expert in the team is added to the hiring cost of the current expert in the loop. In line 12 to 17, we again find and assign the expert having the minimum hiring cost plus communication cost, i.e., has the highest collaboration with other members in the project. For each expert assigned to the temp_team for a particular project we then use our scoring function in line 19, which helps to calculate the temp_profit after all the projects have been assigned needed experts that hold the skill. Line 20 selects the project which returns the maximum profit. The temp_team for that project is selected as the final team for the project. All the experts in the final team are then read one by one where their capacity and hiring cost is adjusted so that the experts selected in the final team are utilized first for remaining projects up till their working capacity is exhausted. This is explained from line 22 to 24. In line 25, the project which has been assigned experts is removed. The iterations are then performed until the conditions in line 2 are violated. The final team along with the remaining budget is returned in line 26.

Cluster Hire algorithm for maximized profit and collaborative team

```
min_cost = MAX_VALUE, budget = 0, temp_team(p) = Ø, min_cost_expert =
0, exp_cost = 0 , highest_profit = 0
While (budget < B and P ≈ Ø) do
  While (p € P) do
    While (s € PSR(p)) do
      While (r € PSR(p)) do
        While(e € E) do
          If e holds the skill s belonging to p && Cap(e) > 0 then
              exp_cost = C(e)
              if ( exp_cost < min_cost and temp_team(p) = Ø) then
                  min_cost_expert = e
                  min_cost = C(e)
              else
                for all e' € temp_team do
                    exp_cost = exp_cost + Dist(e, e')
            if (exp_cost < min_cost)
            min_cost = exp_cost
            min_cost_expert = e
           temp_team(p). add(e, s)
           temp_profit(p) = Revenue(p) – min_cost
   if (temp_profit(p) > highest_profit(p))
   final_team(p) = temp_team(p)
  for all e in final_team(p) do
     budget = budget − C(e)
     update Cap(e), C(e)
     remove p from P
return final_team, budget
```

Figure 5 Heuristic Algorithm for Cluster Hire Problem

3.4.2 Sample data set observations

We now illustrate the output produced by all the three approaches, to see the difference between the decision-making process of the algorithms given the same set of sample input.

Skills: {AI, DB, ML, IR}

Experts: {Expert0, Expert1, Expert2, Expert3}

<u>Social Network amongst the experts:</u>

Expert 0<-> Expert 1, Expert 2

Expert 1 <-> Expert 0, Expert 2

Expert 2<->Expert 1, Expert 3

Expert 3<->Expert 2

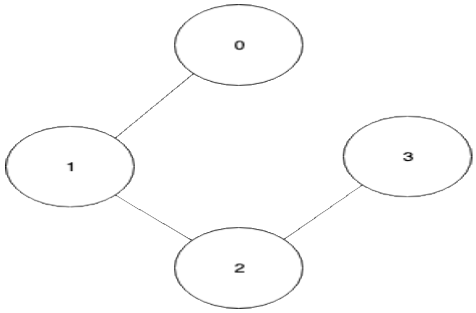

Figure 6 Sample Network of Experts

Given Budget: 500

Projects Covered: Project 1, Project 2, Project 3

Experts Selected: Expert 0, Expert 1, Expert 2, Expert 3

Output from ILP model with maximized profit

Project 1: {AI: Expert 0, DB: Expert 0, Expert 2, ML: Expert 1}

Project 2: {ML: Expert1, Expert 2, IR: Expert 3}

Project 3: {AI: Expert 0, IR: Expert 3}

Budget Exhausted: 0

Output from ILP model with maximized profit and minimized communication cost

Project 1: {AI: Expert 0, DB: Expert 0, Expert 2, ML: Expert 1}

Project 2: {ML: Expert2, Expert 3, IR: Expert 3}

Project 3: {AI: Expert 0, IR: Expert 2}

Budget Exhausted: 0

Expert ID	Expert Skill	Expert Cost	Expert Capacity
Expert0	AI, DB	150	3
Expert1	AI, ML	120	2
Expert2	DB, AI, ML, IR	120	3
Expert3	ML, IR	110	4

Table 2 Expert Attributes

Output from Heuristic Algorithm

Project 2: {ML: Expert1, Expert 2, IR: Expert 2}

Project 3: {AI: Expert 1, IR: Expert 2}

Budget Exhausted: 260

Project ID	Project Skill Demand	Project Profit
Project 1	AI: 1 DB: 2 ML:1	150
Project 2	ML: 2 IR: 1	120
Project 3	IR:1AI:1	150

Table 3 Project Attributes

All the three algorithms provide results as per their design logic. The ILP with maximized profit makes sure to select experts first with less salary and more capacity when their skill matches with the skill required by the project.

The most interesting part is to study how ILP with maximized profit and communication cost and Heuristic Algorithm behave while trying to solve the given problem. The way heuristic algorithm is designed it selects the expert with minimum hiring cost and minimum collaboration cost and tries to exhaust that expert's capacity first whenever a match is found between the project's skill and expert's expertise. While ILP looks through all the solution nodes available to find the experts which has least salary, more capacity, and has the most set of skills needed by the sets of the projects giving the most collaborative team and covering the projects with high revenue amount associated. We observe that expert 2 gets its skill exhausted by heuristic algorithm. Hence, although it has skill *DB*, it is not able to participate to cover the skills and demand of project 1, covering the rest of the project with higher revenue i.e., project 2 and project 3.

CHAPTER 4
RESULT AND ANALYSIS

In this chapter, we evaluate the performance of our proposed algorithms to find the best team of experts for the problem of cluster hiring in a network of experts. The ILP formulation evaluated in this research generates better results which are relatively close to optimal, when compared to heuristic solutions. The primary objective of this thesis is to study and compare the impact of ILP formulation when applied on Cluster Hiring Problem which is NP-Hard in nature.

We formulated two ILP models as described in Chapter 3 to generate solutions for finding a group of experts which satisfy all the constraints provided. The first model produces a group of experts which covers projects which return maximum profit. Second ILP model produces the most collaborative team along with the maximum profit which is formulated by adding additional constraints and objective function used for calculating the communication cost. This gives ILP model a room for flexibility and scalability of the problem domain.

This implies that for each group of expert, the ILP models evaluate

1. The maximum profit that the group of experts return
2. The group of experts selected fall in the feasible region. That is, satisfy all the constraints required for the problem solution.

The objective of the model is that the total profit should be the maximum and the team formed should be the most collaborative.

We have used Gurobi Optimizer, a mathematical solver written in python to form the model for the selection of a group of individuals that can help complete a subset of projects in a given budget. Gurobi Optimizer is a commercial optimization solver which is used to solve linear optimization problems known as linear programming(LP), including Integer Linear Programming (ILP). Guorbi Optimizer provides flexible, high-performance mathematical programming solver for linear programming(LP), quadratic programming (QP) problem, quadratically constrained programming(QCP) and mixed integer programming(MIP) problems [12].

For result analysis, we also developed a Heuristic algorithm as described in chapter 3 for comparison with ILP solution. Below are the inputs to both the ILP model and our Heuristic algorithm.

1. A file containing the network of experts with edge weights representing the co-authorship between two experts
2. A file containing the expert's cost and capacity information along with the list of skills he has expertise in
3. A file consisting of set of projects. Each project consists of

   a) project revenue

   b) Skills required for the project

   c) Number of experts required per skill by a project

We form our ILP models for the cluster hire problem with objective function and constraints - to meet all the criteria required to form a group of experts. The execution of the ILP solver for our model takes following steps.

1.Reads the project, the skills required by the project and the number of experts needed for each skill

2. Format constraints for our ILP model so that the selected experts meet the objective of the team formation. These constraints take into account expert capacity, expert cost, expert skills, communication cost between the experts from the input files, and are saved in a file of type lp.

3. Invokes Gurobi solver. The solver reads the .Lp file and attempts to find, if possible, an optimal solution for the problem.

4. The ILP solver first finds the best solution for the given problem without considering the constraints. This helps define the upper bound for the given maximization problem.For example, the ILP solver first considers the least communication cost value and the maximum profit value that can be obtained by selecting the most colloborating group of experts that covers all the given set of projects/skills.

5. The solver then finds the best solution that gives a value near the upper bound and fullfils all the conditions formulated to solve the problem.In this scenario, the ILP solver in each step forms groups of experts that satisfy all the set of constraints required for our cluster hire problem and checks if the value returned is less than or equal to the value obtained in Step 4 to return the group of experts and projects covered.

The ILP model formulated to solve cluster hire problem is tested against varied size of data sets which generates five binary variables, each consisting of a different combination of numbers of experts k, a list of skills s and projects p.

The size of the decision variable denoting which experts is selected for which skill of which project (Viup) is k*m*n.

Similarly, the number of decision variables for calculating the collaboration between experts of the same project (Ziujvp) is k*$m^2 * n^2$

For instance, let’s assume m=8 experts, n=3 skills and k is 2 projects. Then below will be the size of the above described variables

Expert-Skill selected for project(Viup) will be 8 * 3 * 2 = 48

Expert-Skill-Expert-Skill for project to get communication cost(Ziujvp) will be 8 * 3* 8 * 3 * 2 = 4608

Hence, the size of the variables increases with the increase in the size of the dataset, producing memory limitation for loading the model to the memory and solving the problem. Thus, for our experimentation, we run the cluster hire ILP model with maximized profit on 10k nodes while we limit the ILP model with maximized profit and minimized communication cost to 50 nodes of experts given the memory limitation of the hardware used.

The following are the ILP models with their objectives, how many nodes of experts are used in this study for experimentation, and the number of variables required by the model for a different collection of projects:

ILP Model	No. of nodes	No of Skills	No. of variables	Size of the lp file
ILP model for cluster hiring with maximized profit	10,000 nodes	59	10 Projects-5,900,000 20Projects-11800000	10 Projects-0.69 GB 20 Projects-1.41 GB
ILP model for cluster hiring with maximized profit and most collaborative	50 nodes	26	05 Projects-16,906,500 08 Projects-27,050,400	05 Projects -4.31GB 08 Projects-7.38GB

Table 4 Variable Size

To obtain the dataset, we first create the input graph (an, i.e., network of experts) from the DBLP. The dataset consists of the information about a set of papers and their authors. It consists of information about 10,000 experts. The edges represent the collaboration or co-authorship information between the researchers. The second dataset is a random network of 50 Nodes with edge weights applied to represent the collaboration similar to that in DBLP but with random numbers. Note that for our experiments, edge weights (here collaboration), expert's costs, expert's capacity, project profit, skill demand for experts are normalized to have the same scale. We randomly create a collection of projects, each collection consisting of 5, 8, 10 and 20 projects using the below scales for our two networks. In our experiments, we use different values for the budget to see the total profit returned by each technique.

Synthetic Data generation and Scales used

Dataset	Expert's Cost	Expert's Capacity	Project's Revenue	Skill demand of projects
10,000 Nodes	100 to 700	5 to 8	100 to 200	4 to 9
50 Nodes	100 to 200	1 to 3	50 to 100	1 to 3

Table 5 Scale of values used

We study the results achieved using both ILP and Heuristic approaches. Both techniques have their own approach of selecting a group of experts which tend to complete the given set of the projects within the required budget. We compare the results of ILP implemented in python and Heuristic algorithm implemented in Java on Intel Core i7 3.3 GHz computer with 32GB of RAM.

4.1 Heuristic vs Cluster Hire Profit Model (10,000 Nodes)

4.1.1 Total Profit vs Budget

We present the experimental results in this section. We studied how efficiently the ILP model produces profit compared to the heuristic algorithm. We evaluate the effect of the budget on the total profit of projects. The figure below shows the values of the total profit for increasing values of the budget for a different number of projects. Here, for comparison with the heuristic algorithm, we have considered k projects, where k= {10,20}. We then calculate the total profit for different values of the budget. The results suggest that our ILP model achieved a higher profit compared to the heuristic algorithm when we have a limited budget. The ILP model covers more profitable project as the budget increases.

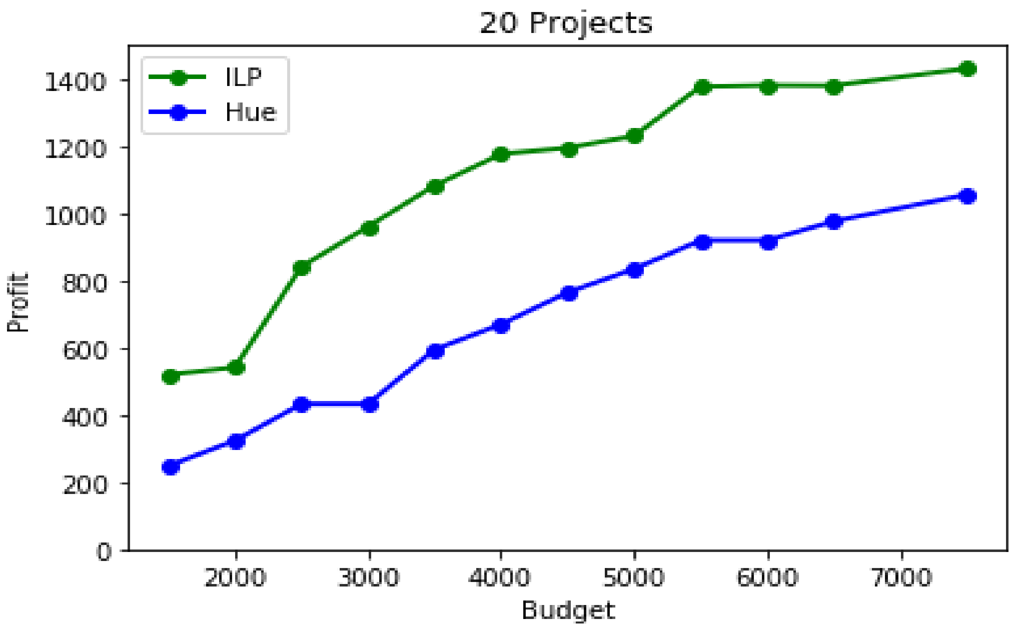

Figure 7 Comparison between budge and profit with project size 20

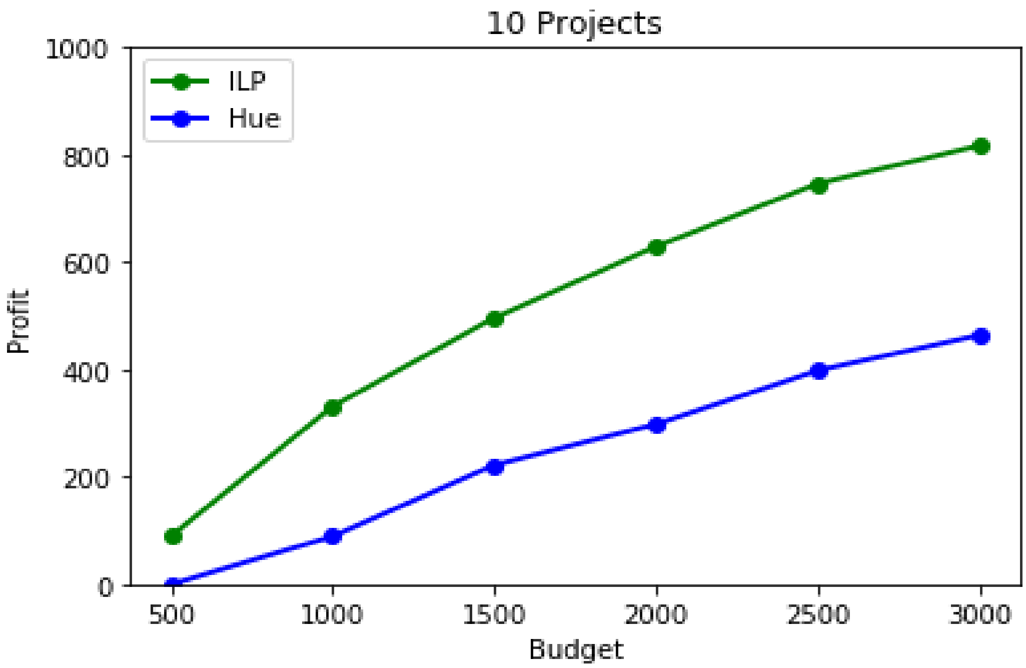

Figure 8 Comparison between budge and profit with project size 10

4.1.2 Number of Projects completed vs Budget

The figure below represents the number of completed projects while increasing the value of the budget for a different number of projects. The results show the efficiency of the ILP outperforming that of the heuristic solution even when we have a limited budget. For higher values of the budget, we can see that the heuristic solution gives less number of projects compared to the ILP technique.

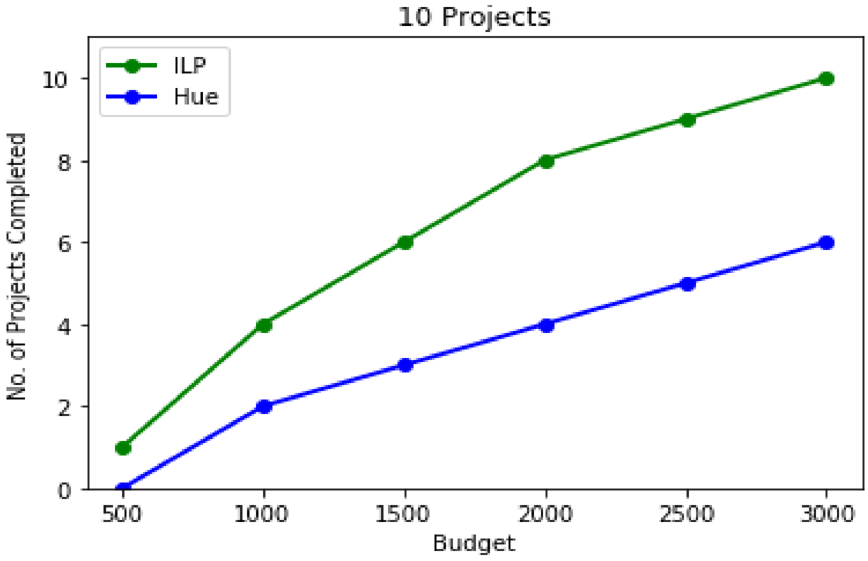

Figure 9 Number of completed projects vs budget (10 projects)

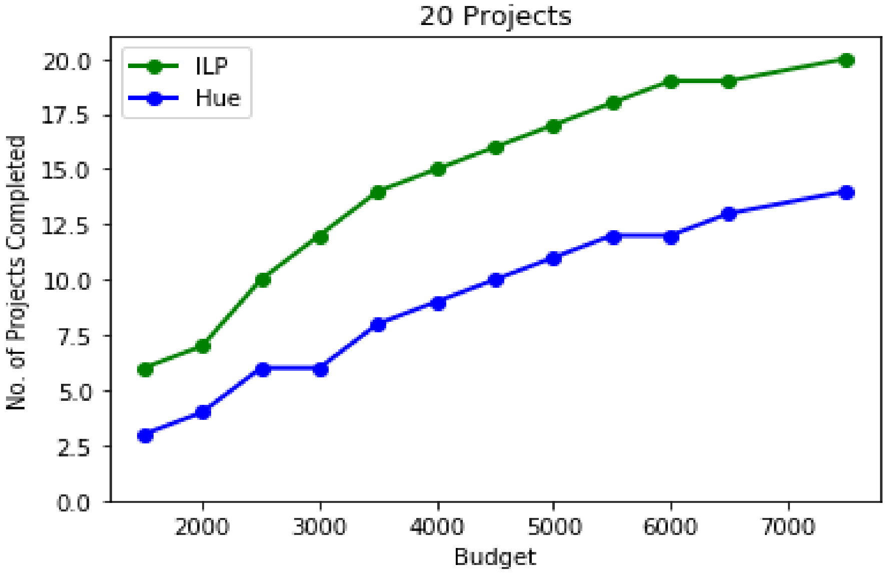

Figure 10 Number of completed projects vs budget (20 projects)

4.2 Heuristic vs Cluster Hire Profit and Communication Cost Model (50 Nodes)

4.2.1 Total Communication Cost vs Budget

We now study the effect of the budget on communication cost with a comparison between the ILP model producing the maximum profit with the most collaborative team and heuristic algorithm having the same objective. We experimented on 50 nodes of experts and two sets of projects p. Each set of project as discussed before consists of information regarding which skills are required for the project development and the number of personnel needed for the project's successful completion. We observe from the below graph that initially both the heuristic and ILP model result in the almost similar amount of communication cost, but as the budget increases, ILP tends to produce the most collaborative team while the heuristic one constantly increases the sum of communication cost between the experts selected for each project completed. ILP produces the minimum communication cost and hence produces the most collaborative team.

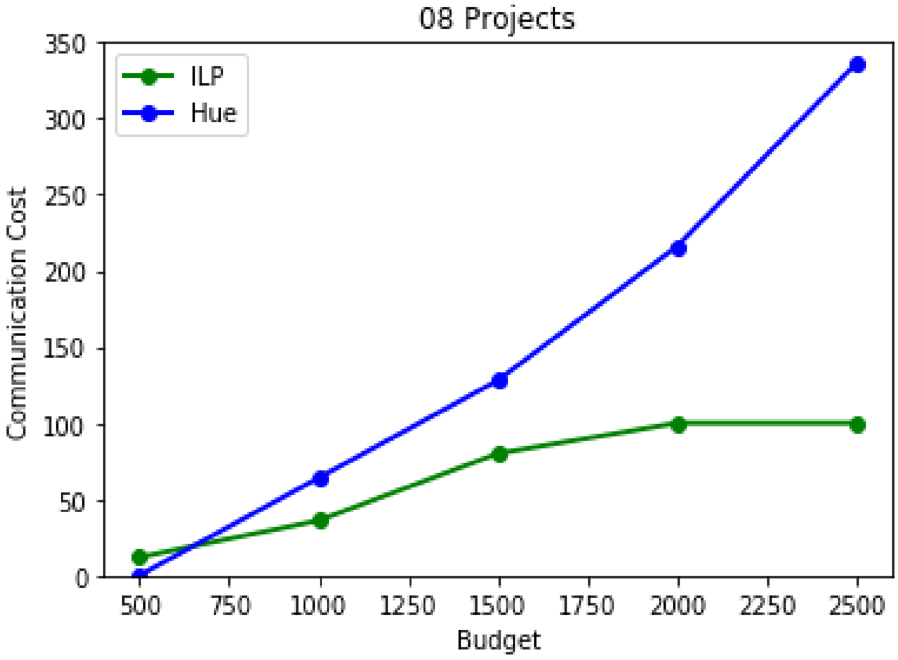

Figure 11 Communication cost vs Budget (08 Projects)

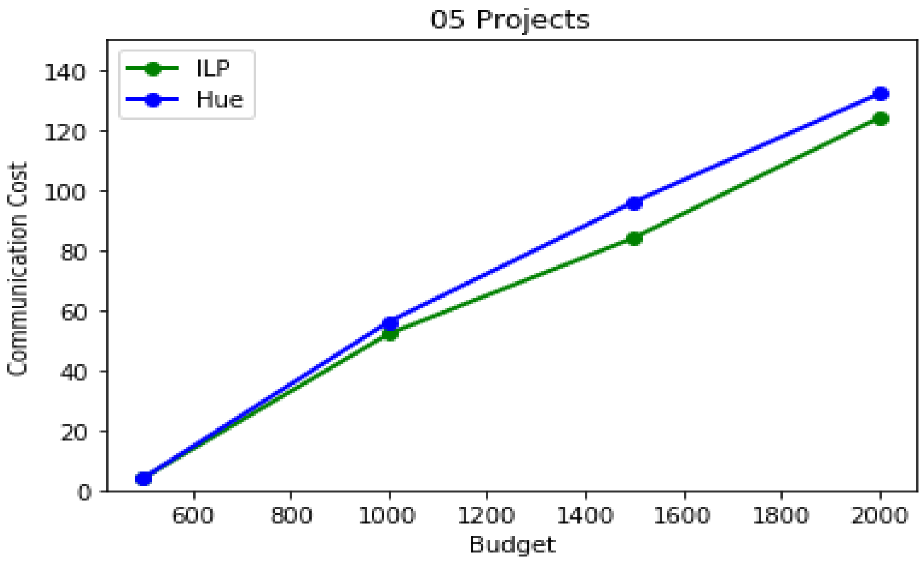

Figure 12 Communication cost vs Budget (05 Projects)

4.2.2 Total Profit vs Budget

Like above, we now compare the maximum profit obtained by ILP with the heuristic algorithm to see the effect of increasing budget on total profit earned. We notice that the heuristic algorithm produces close to optimal results as produced by ILP but as more projects are included, the profit returned by the ILP model increases with large leaps.

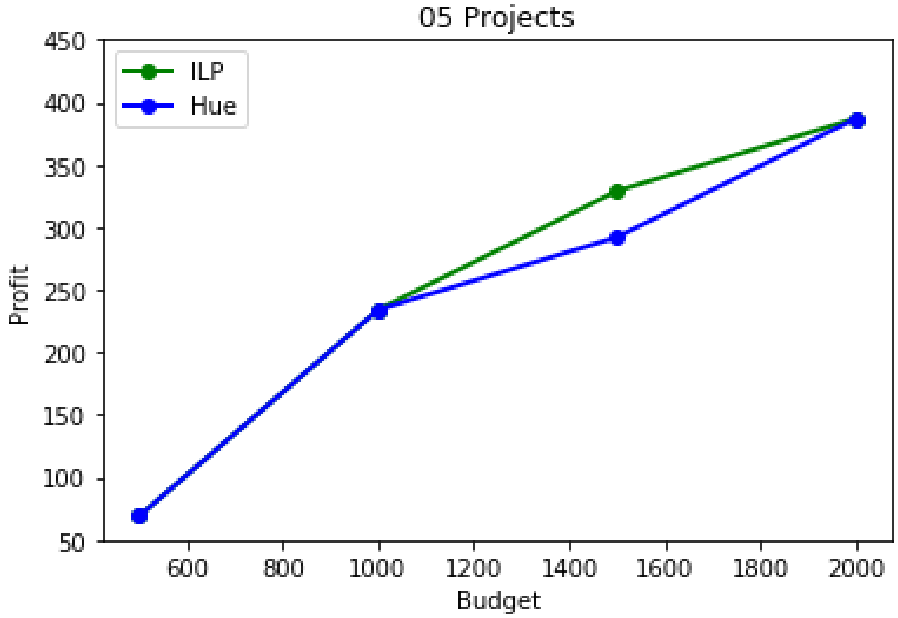

Figure 13 Profit vs Budget (05 Projects)

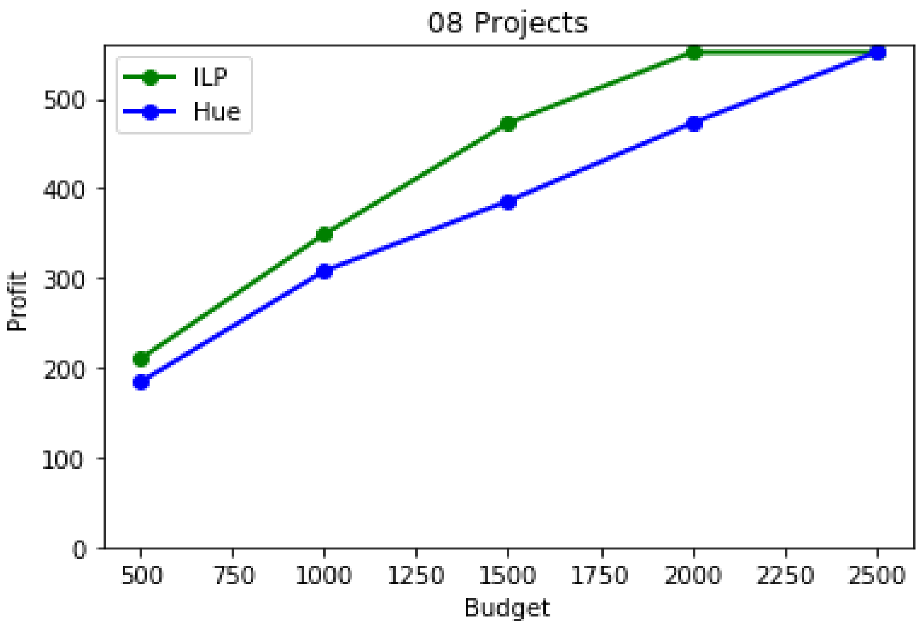

Figure 14 Profit vs Budget (08 Projects)

4.3 Discussion

In this section, we will discuss the performance of our proposed algorithms to solve the cluster hire problem with an objective to select group of experts which bring minimal profit and minimal communication cost when there is collaboration amongst the project members. We present two ILP models to solve the Cluster Hire problem, each having its own objective. We performed various experiments and compared the results with heuristic algorithm to evaluate the performance of our approach.

Using DBLP data set, we compare the ILP model having profit maximization objective with Heuristic algorithm. We perform experiments that compare the total profit with the budget and number of completed projects with budget. We perform the comparison of total profit vs budget with two set of projects having size 10 and 20 respectively. The results show that solving the cluster hire problem by formulating it in form of an ILP model gives better results as compared to heuristic technique. One of the primary reason is the way the ILP solver works. While the heuristic solution behaves in the way it is designed, that is, picking up the expert with the least salary and communication cost for the project, the ILP solver goes through various combinations of group of experts which returns the most profitable subset of projects. Since the ILP solver covers all the best possible combinations out of available experts, skills and projects, it takes more execution time compared to the Heuristic algorithm. For example, heuristic solution takes execution time of 11.45 seconds while the ILP solver takes approximately 6 minutes to return the group of experts for the selected project constrained by the given budget. A similar trend is observed when the same experiments is performed by varying number of projects. When considering an experiment which compares the number of completed projects with the budget, we observe that the ILP technique covers more projects as compared to the heuristic algorithm.

After discussing the experiments with DBLP dataset, we now discuss the experiments performed using Synthetic Dataset where we compare the ILP model having the objective to maximized both profit and collaboration amongst the project members with the same heuristic solution we used above. We first compare the total communication cost vs budget. Note the higher the value of the communication cost, the less the collaboration amongst the project members. Again, here we notice that both the ILP and heuristic solution produce good collaborative teams when there is small number of projects, but as we increase the project size we can see notable difference. The ILP approach produces more collaborative team in contrast to the heuristic approach. We notice that the value of the communication cost keeps increasing with the increase in the budget. After this experiment, we then consider the total profit vs budget comparison. We observe that for small number of projects, the value of the total profit earned is almost the same as the one returned by the ILP solver. When the project number is increased, the ILP solver outperforms the heuristic algorithm

From all the above experiments carried, we find that the ILP models for cluster hiring tends to produce better results. However, it comes with two limitations. First, the size of the variables increases with the increase in the size of the data set. In general, if we increase the number of skills required by the projects, the size of the variables which consists of the expert-skill per project increases drastically, since it will consider all the possible combination of each expert with that skill. Since variables of the ILP model we created are first loaded into the memory, with the increase in the data size we require more memory. Hence, we provide varied problem sizes to check the maximum size of input that can be provided, given the memory limitation as shown below in Fig 15.

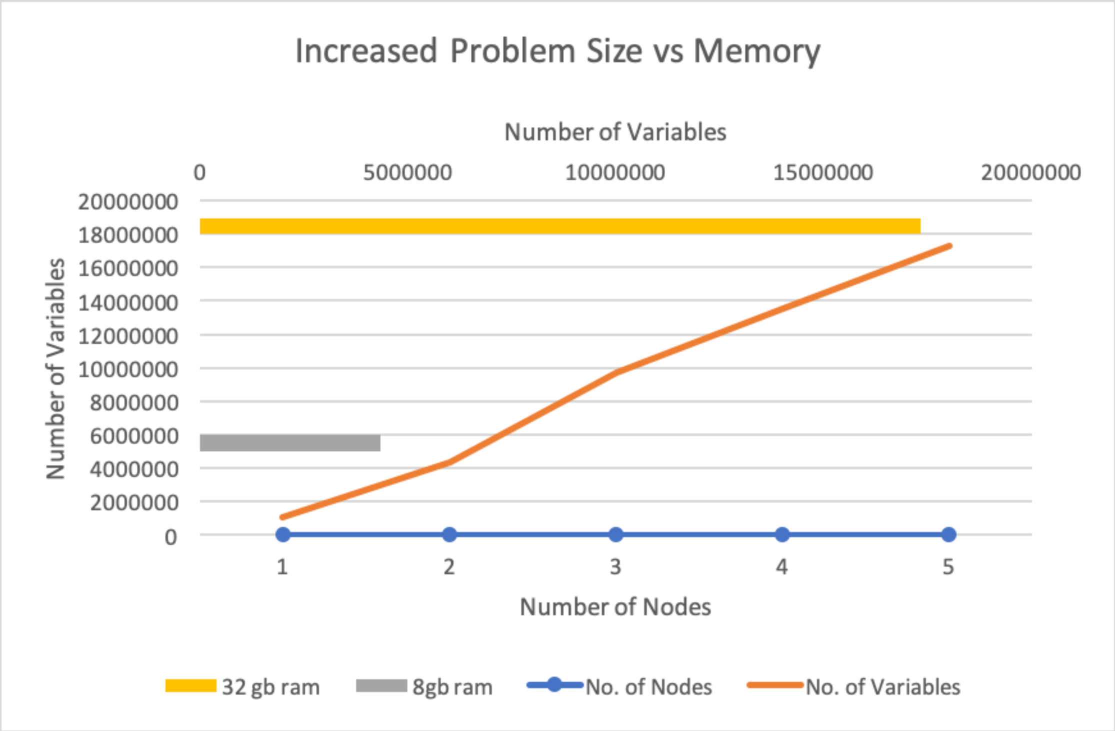

Figure 15 Problem Size Vs Memory Size

We observe that 8GB memory machine can solve the problem, for example with the variable size of. Similarly, with 32GB RAM, the maximum input size can be of 50 nodes with 26 skills and 8 projects respectively. Exceeding the input size of the above raises “Out of memory” error.

The second limitation is the running time of the ILP approach. The cluster hiring problem is an NP-Hard problem, which implies that any optimal algorithm including the ILP algorithms, will likely be time consuming as the problem size becomes large. To evaluate the scalability of the ILP approach on the formulated cluster hiring problem, we experimented with different problem sizes. The runtime results are shown in Fig 16.

As we can observe from the figure, the computation time is fairly short when the problem size is small. The first to experiments each finished within a few seconds to a minute. However, when the problem size increases, the runtime increases rapidly. The last experiment was carried on ILP model with profit maximization and minimized communication cost for an input size of 50 experts ,26 skills and 8 projects. It took minimum 360 seconds approximately to return the group of experts. Overall, with the way we set up our models for cluster hiring, the ILP approach looks favorable on small problem sizes which require less number of variables

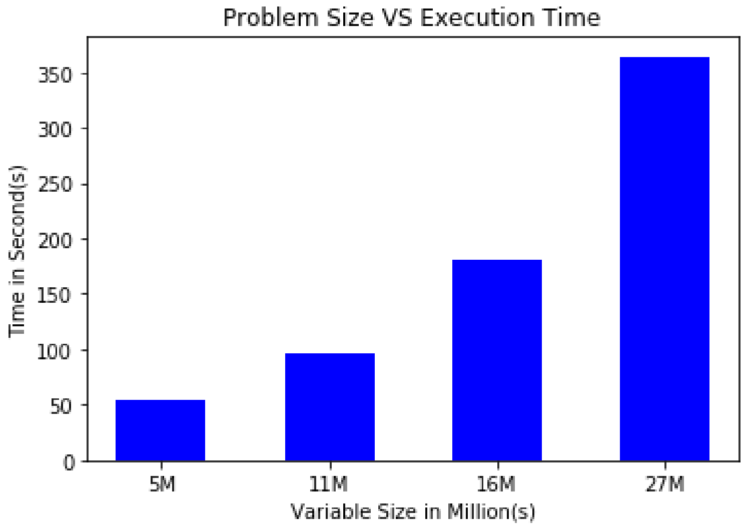

Figure 16 Problem Size VS Execution Time

The goal of this study was to explore the suitability of using ILP formulations and algorithms to solve cluster hiring problem with varied objectives. As explained throughout the thesis, such problems can be very challenging since they involve many difficult constraints. It is not easy to develop good heuristic algorithms for solving such problems. Known heuristic algorithms generally work on simpler versions of the problems and even in those cases, often fail to find feasible solutions and/or under-achieving in terms of the optimizing objective.

From this study, we can conclude that for problems up to certain size, the ILP approach is entirely applicable. With ILP, one can describe the problem precisely and then resort to ILP algorithms to find not only feasible but optimal solutions. For use cases where the problems are solvable by ILP, there is little need to look for heuristic algorithms. It is only when the use cases generate large problems that heuristic algorithms become useful.

CHAPTER 5
CONCLUSION AND FUTURE WORK

5.1 Conclusion

There has been much research done recently to address the Cluster Hire problem, where we want to hire a group of experts to complete a subset of projects within a given budget. Throughout this study, the emphasis was the role of Integer Linear Programming in solving the Cluster Hire problem. According to the experimental results, solving the cluster hire problem using mathematical programming approach has a remarkable impact on finding optimal/near optimal group of experts, and is significantly effective compared to the previously proposed heuristic solutions.

In addition, we used a more practical setting of the cluster hire problem, derived from a more realistic scenario of hiring and can be applied to almost all departments where recruitment of individuals with expertise in various kills is required. We consider a group of experts to be hired for a set of projects each of which has its skills and each skill has its demand asking for how many experts to be hired per skill of the project.

For comparison with the ILP solutions, we also implemented heuristic solution, which helps bring a distinction in the quality of results obtained applying both of the approaches. Our main objectives here are to maximize the profit obtained from the revenue of the projects covered and to minimize the communication cost between the experts to form a collaborative team. Our results from the ILP models conclude that it returns maximum profit and groups experts which have higher collaboration amongst them when compared to heuristic solution. The ILP solution takes considerable amount of memory for variable and constraint creation. Also, since it traverses through all possible best objective functions, it takes longer execution time to return a solution as compared to the heuristic solution.

5.2 Future Work

1) Computational difficulties in solving problems using Integer Linear Programming(ILP) are caused by a considerable degree by the number of variables. If the numbers are small, then even complex problems usually can be solved with

a reasonable expenditure of effort. One of the future work could be to consider to reduce the number of variables. This shall help cope with the computational limitations faced in this study.

2) In future work, we may also consider extending this problem by adding some additional features to the cluster hiring problem. We can consider work experience as one of the factors affecting the selection of the individuals for the project assignment.

REFERENCES/BIBLIOGRAPHY

1. 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). ACM.
2. 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). ACM.
3. Gaston, M., Simmons, J., & DesJardins, M. (2004, July). Adapting network structure for efficient team formation. In Proceedings of the AAAI 2004 fall symposium on artificial multi-agent learning.
4. 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). ACM.
5. Wi, H., Oh, S., Mun, J., & Jung, M. (2012). A team formation model based on knowledge and collaboration. IEEE Engineering Management Review, 1(40), 44-57.
6. Data mining - knowledge discovery: www.tutorialspoint.com/data mining/dm knowledge discovery.htm
7. Baykasoglu, A., Dereli, T., & Das, S. (2007). Project team selection using fuzzy optimization approach. Cybernetics and Systems: An International Journal, 38(2), 155-185.
8. http://web.tecnico.ulisboa.pt/mcasquilho/compute/_linpro/TaylorB_module_c.pdf
9. Fitzpatrick, E. L., & Askin, R. G. (2005). Forming effective worker teams with multi-functional skill requirements. Computers & Industrial Engineering, 48(3), 593-608.
10. 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). ACM.
11. https://www.doc.ic.ac.uk/~br/berc/integerprog.pdf
12. https://en.wikipedia.org/wiki/Gurobi

13. 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). ACM.
14. Patel, M. S. (2018). Cluster Hire in a Network of Experts.
15. http://www.mi.fuberlin.de/wiki/pub/Main/GunnarKlauP1winter0708/discMath_klau_ILP_I.pdf
16. X. Wang, Z. Zhao, and W. Ng, "USTF: A Unified System of Team Formation," IEEE Transactions on Big Data, vol. 2, no. 1, 2016.
17. 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.
18. http://www3.imperial.ac.uk/pls/portallive/docs/1/7527705.PDF

APPENDICES

Appendix A

Sample code for ILP with a simple example

```
# -*- coding: utf-8 -*-
"""
Created on Sun Mar 17 15:58:30 2019

@author: Sagarika
"""

import gurobipy  as grb
GRB = grb.GRB
import pandas as pd


Expert_Skill={

         (0,"AI"):1,(0,"DB"):1,(0,"ML"):0,(0,"Python"):0,(0,"IR"):0,
         (1,"AI"):1,(1,"DB"):1,(1,"ML"):1,(1,"Python"):0,(1,"IR"):0,
         (2,"AI"):1,(2,"DB"):1,(2,"ML"):1,(2,"Python"):0,(2,"IR"):1,
         (3,"AI"):1,(3,"DB"):0,(3,"ML"):1,(3,"Python"):0,(3,"IR"):1,
        }


expert_costs={0:150,1:120,2:120,3:110,4:200}

expert_capacity={0:3,1:3,2:3,3:3,4:4}

Project_Skills={
1:{"AI":2,"DB":2,"ML":2},2:{"ML":1,"IR":1},3:{"IR":1,"AI":1} }

Project_Profit={1:120,2:110,3:150}

projects=[1,2,3]

Expert_Communication_Cost=[
    [0,5,2,2],
    [5,0,5,2],
    [2,5,0,5],
    [2,2,5,0]
  ]


Experts=[0,1,2,3]

Skills=[ "AI","DB","ML","IR","Python"]


m=grb.Model("Cluster Hirinig")
```

```
Viup= m.addVars(Experts,Skills,projects,vtype=GRB.BINARY,name="V")

z=
m.addVars(Experts,Skills,Experts,Skills,projects,vtype=GRB.BINARY,na
me="Z")

y=m.addVars(Experts,Skills,Experts,Skills,projects,vtype=GRB.BINARY,
name="Y")

Wvals=[(p) for p,val in Project_Skills.items()]
Wp=m.addVars(Wvals,vtype=GRB.BINARY,name="Wp")


WVals=[(e) for e in Experts]
Wi =m.addVars(WVals,vtype=GRB.BINARY,name="Wi")



m.setObjective(sum(Wp[(project)] * Project_Profit.get(project) for
project in projects) -
sum(y[e,s,e1,s1,p]*(Expert_Communication_Cost[e][e1]) for e in
Experts for s in Skills for p in projects for e1 in Experts  for s1
in Skills),GRB.MAXIMIZE)

for e in Experts:
    for s in Skills:
          m.addConstr( sum(Viup[e,s,p] for p in projects) <=
Expert_Skill[(e,s)] )


for project_no,skills in Project_Skills.items():
    for skill,req in skills.items():
       m.addConstr( sum(Viup[expert,skill,project_no] for expert in
Experts) >= Project_Skills.get(project_no,{}).get(skill) *
Wp[(project_no)] )


for e in Experts:
            m.addConstr( sum(Viup[e,s,p] for p in projects for s in
Skills) <= expert_capacity.get(e) )


for p in projects:
    for s in Skills:
        for e in Experts:
            m.addConstr( (Wi[(e)]) >= (Viup[e,s,p]) )

for e in Experts:
m.addConstr(  (Wi[(e)]) <=  sum(Viup[e,s,p] for p in projects for s
in Skills)   )

c =m.addConstr(  sum(Wi[(e)] * expert_costs.get(e) for e in Experts)
<= 270 )
c.RHS=700
```

```
#z_{i_u_p_j_v_p} <= V_i_u_p + V_i_u_p -1

for p in projects:
    for u in Skills:
        for v in Skills:
            for i in Experts:
                for j in Experts:

                            m.addConstr( z[i,u,j,v,p] <= Viup[i,u,p]
)
                            m.addConstr( z[i,u,j,v,p] <= Viup[j,v,p]
)
                            m.addConstr( z[i,u,j,v,p] >= Viup[i,u,p]
+ Viup[j,v,p] -1 )
                            m.addConstr( y[i,u,j,v,p]  <=
z[i,u,j,v,p] )
                            m.addConstr( y[i,u,j,v,p]  <= Wp[p] )
                            m.addConstr( y[i,u,j,v,p]  >=
z[i,u,j,v,p]+ Wp[p] - 1 )


c.RHS=110
m.setParam("Timelimit",100)
m.write('ILP.lp')

m.optimize()

for v in m.getVars():
        if v.x!=0.0:
            print(v.varName, v.x)

print('Obj:', m.objVal)
```

VITA AUCTORIS

NAME:	Sagarika Khandelwal
PLACE OF BIRTH:	Dombivali, Maharashtra, India
YEAR OF BIRTH:	1993
EDUCATION:	Auxilium Convent High School, Gujarat, India 2009 Parul Polytechnic Institute, Diploma, Gujarat, India 2012 L.D. Institute of Engineering, Gujarat, India, 2015 University of Windsor, M.Sc., Windsor, ON,Canada 2019