M.Sc. Engg. Thesis

AN AHP AND TOPSIS BASED APPROACH TO SOLVE STUDENT-SUPERVISOR ASSIGNMENT PROBLEM

by

Tamanna Tasneem (1014052057 P)

Submitted to

Department of Computer Science & Engineering

(In partial fulfillment of the requirements for the degree of Master of Science in Computer Science & Engineering)

Department of Computer Science & Engineering

Bangladesh University of Engineering & Technology (BUET)

Dhaka 1000

November 30, 2021

Dedicated to my loving parents and supervisor madam

AUTHOR'S CONTACT

Tamanna Tasneem

Email: tamannacse09@gmail.com

The thesis titled "AN AHP AND TOPSIS BASED APPROACH TO SOLVE STUDENT-SUPERVISOR ASSIGNMENT PROBLEM", submitted by Tamanna Tasneem, Roll No. **1014052057 P**, Session October 2014, to the Department of Computer Science & Engineering, Bangladesh University of Engineering & Technology, has been accepted as satisfactory in partial fulfillment of the requirements for the degree of Master of Science in Computer Science & Engineering and approved as to its style and contents. Examination held on November 20, 2021.

Board of Examiners

1. SSharmin ______________________

Dr. Sadia Sharmin
Assistant Professor
Department of Computer Science & Engineering
Bangladesh University of Engineering & Technology, Dhaka.

Chairman
(Supervisor)

2. ______________________

Prof. Dr. A.K.M. Ashikur Rahman
Head and Professor
Department of Computer Science & Engineering
Bangladesh University of Engineering & Technology, Dhaka.

Member
(Ex-Officio)

3. ______________________

Prof. Dr. Mahmuda Naznin
Professor
Department of Computer Science & Engineering
Bangladesh University of Engineering & Technology, Dhaka.

Member

4. ______________________

Dr. Atif Hasan Rahman
Assistant Professor
Department of Computer Science & Engineering
Bangladesh University of Engineering & Technology, Dhaka.

Member

5. ______________________

Prof. Dr. M. Kaykobad
Distinguished Professor
Department of Computer Science & Engineering
Brac University, Dhaka.

Member
(External)

Candidate's Declaration

This is hereby declared that the work titled "AN AHP AND TOPSIS BASED APPROACH TO SOLVE STUDENT-SUPERVISOR ASSIGNMENT PROBLEM", is the outcome of research carried out by me under the supervision of Dr. Sadia Sharmin, in the Department of Computer Science & Engineering, Bangladesh University of Engineering & Technology, Dhaka 1000. It is also declared that this thesis or any part of it has not been submitted elsewhere for the award of any degree or diploma.

Tamanna Tasneem

Tamanna Tasneem

Candidate

Acknowledgment

First of all, I would like to express my heart-felt gratitude to my supervisor, Dr. Sadia Sharmin, for her constant supervision of this work. And secondly, to my parents and family members, who supported me throughout this journey.

Abstract

The growing education sector of the current era offers a lot of choices and different paths for the students as well as for the educators. They now have a lot more opportunities to work with someone who has specific skills or particular fields of interests. At the same time, more options poses more challenges for the students and educators while narrowing down the list of potential fellows. Finding a desired match from this large pool of students and supervisors/grad-schools, where both student and supervisor have different preferred criteria, is a mammoth task. Our study proposes a way to cut down this list to a smaller size and help both the parties in finding a suitable match. The solution we suggest is, ranking the students and supervisors using AHP and TOPSIS methods according to each of their requirements, preferences and qualities. This ultimately produces a customized list of potential matches for each of the students and supervisors and the problem becomes a linear single-objective optimization problem. From this ranked list we then pair the students with supervisors while taking into account the satisfaction of both parties about the matching. The major contribution of this work is it suggests a way to match pairs based on their criteria rather than over the participants themselves. Finally, we provide the results of our solution that proves the effectiveness of our model above other solutions which only ensures the satisfaction of one of the parties involved in such a match and also discuss the results with comparison to algorithms such as Genetic Algorithm which takes into account both parties satisfaction.

Acronyms List

MOEA = Multi-Objective Evolutionary Algorithm

MOEA/D = Multi-Objective Evolutionary Algorithm based on Decomposition

MOP = Multi-Objective Optimization Problem

NSGA = Non-dominated Sorting Genetic Algorithm

SMP = Stable Matching Problem

NRMP = National Resident Matching Program

GPA = Grade Point Average

CGPA = Cumulative Grade Point Average

AHP = Analytical Hierarchy Process

TOPSIS = Technique for Order of Preference by Similarity to Ideal Solution

SAP = Sailor Assignment Problem

SPA = Student Project Allocation

Contents

List of Figures

Chapter 1

Introduction

The challenges faced by students while finding suitable supervisors during grad/post-grad research are tenacious. The growing education sector these days offers a lot of choices and different paths for the students as well as to the supervisors to work with specific skills or interests [2]. Finding a suitable match for grad/post-grad students is very important for the success of the research. Studies indicate that the supervisor-doctoral student interpersonal relationship is important for the success of a PhD-project [2, 3]. Ives and Rowley for example reported that good interpersonal working relationships between supervisors and their PhD students were associated with good progress and student satisfaction [4]. But to find a suitable match from the large pool of students and supervisors/grad-schools, where student and supervisor both has different preferred criteria is a difficult task.

Until now the selection process of both students and supervisors/grad-schools by the supervisors/grad-schools and students respectively has remain a free-for-all market where each student and supervisor/grad-school are able to negotiate with one another directly in order to find a suitable match. But there are several problems pointed out by the economists in free-for-all markets. [5, 6]

For example, the problem of *unravelling* in which the related parties may form assignments with one another in advance and earlier than the deadline of forming all assignments. For example, grad-schools wishing to enroll the best students might compete with one another by advancing the date when they make their offers. This may lead to students settling for a supervisor/grad-school which wasn't one of their top priorities.

To avoid this issue, students might be prevented from entering into premature assignments before a certain date. But this again can lead to the problem of *congestion*, in which all the students and supervisors/grad-schools do not have sufficient time to negotiate with one another over potential matching prior to the deadline. For example a grad-school *G* offering 20 places to applicants might have to make substantially more than 20 offers, to allow for applicants who will turn down *G*'s offer.

Again while trying avoid congestion, a new problem may arise, namely *exploding offers*. In such a case, students are given only a short time period to decide whether they will be able to accept the offer, otherwise the opportunity for accepting the offer is removed. For example grad-school *G* might force an applicant *A* to make a decision swiftly by setting *A* a deadline, beyond which the offer expires. These problems of Unravelling, congestion and exploding offers may lead to a situation where the students and the supervisors/grad-schools are forced into forming associations with one another before they have enough knowledge about the whole range of possible assignments that may be available to them.

But on the other hand, it is also very difficult to zero in on suitable supervisors who the student can approach when they don't know the supervisors' personally. And to look up each and every one of them will be like climbing a hill. It'd be easier if they have a short or narrow list available to them according to their choice and preference in topic, grad schools, supervisors' background etc.

The same is true for the supervisors too. When a supervisor has to accept students to work with them they have their own sorting criteria. The qualities they look for a student like their CGPA, experience with research etc. So similarly with a similar personalized list for them will be very helpful in this task.

Centralised matching schemes (referred to as (centralised) clearinghouses by economists) can avoid some of the problems that are inherent in free-for-all markets. These works along the following lines: the input data involving the agents and their preferences over one another are collected by a given deadline by a trusted central authority. This third party in turn creates an optimal matching with respect to the supplied preference lists and capacities, and any other problem-specific constraints. By participating in the process, the parties have to agree to the matching outcomes created by it. The exact definition of an optimal matching may have many variations depending on the context, for example, it might mean maximising the number of places that are filled at each hospital, or giving the maximum number of school-leavers their first-choice university, or ensuring that no junior doctor and hospital have an incentive to reject their assignee's and become matched to each-other, if they do not prefer their current partners.

Centralised matching schemes are often given a name that is a term that can be used for the entire administrative and algorithmic process of data collection, computation of a matching and publication of the outcome. For example the assignment of junior doctors to hospitals in the US is handled by the National Resident1 Matching Program (NRMP) [7, 8]. In 2012, 38,777 aspiring residents applied via the NRMP for 26,772 available residency positions [8].

Current Trends

Higher studies have become a much sought out aspect during the last three to four decades. Percentage of the students who are aiming for a higher degree after under-graduation is increasing day by day. In 2011, there were 730,635 master's degrees conferred by degree-granting institutions compared to 463,185 in 2000 (NCES, 2011a). So finding out what they are seeking in a master's program is becoming increasingly important. There have been multiple factors contributing to this growing trend -

Invest in your future Although all the students may not have a clear view of what they want to do in the future before applying to graduate schools, it'll certainly help to have at-least an idea. This is because grad schools can often be considered as a professional training that enables and ensures students to graduate with all the required knowledge for the appropriate jobs so that they are ready to jump straight into their desired careers.

job and financial prospects As more and more people are attending graduate school today, an undergraduate degree alone can sometimes fail to get a person noticed when there are equally or more highly qualified candidates. Also a down economy encourages students to pursue graduate education instead of entering the job market [9] and an increase in jobs that require a post- baccalaureate degree. For example, in the UK, university education is increasingly being viewed as more of a necessity than a luxury, and 11% of the workforce there now holds a graduate degree. As a result, bachelor degree holders are struggling to appeal to employers even at entry level in certain industries – especially when up against candidates with

PhDs.

Bettered financial prospects are also a popular answer to the question "Why go to graduate school?" – though it may not necessarily be the most important factor but it still helps to find better financial prospects in job sector with a recognized degree.

Get more than a qualification Whereas much of the worth of an undergraduate degree is in the qualification itself, the most important reasons to go to grad school may be more for the professional skills you'll gain, the personal development you'll achieve and the helpful connections you'll make with other graduate students, academics and industry experts.

Pursue your interests in more depth and Contribute to the world's knowledge Although a student gets the opportunity to study specific topics and take classes of personal interest, a graduate degree helps to do the same to a much greater extent. In order to get the most out of your graduate degree, you will be expected to conduct personal research alongside set study topics, in order to develop your thoughts and ideas regarding something that deeply interests you.

Also if you're someone who wants to contribute to the world within any field, professionally or academically, you're going to have to know your subject inside-out. For STEM (Science, Technology, Engineering and Mathematics) subjects or other highly specialized fields, grad school helps to make that happen.

Make connections Grad school is much more about connecting with people professionally – not just fellow graduate students but faculty members too. A graduate degree student needs to learn how to reach out and make connection with people.

Due to these factors present day students are getting more and more inclined to choose higher degrees and one of the first steps of finding this is to find a suitable supervisor.

Background

Generally a student when searching for a supervisor/grad-school has to go through his/her options manually. They collect data from different sources like previous students, their current faculty members, online search or student agencies. Then from these data they try to pin point one or two supervisors or grad schools that matches their preferences more. In the undergrad level where the students know their perspective supervisors it is a whole lot easier to choose from them. Different algorithmic approach has been applied to help this assignment though the years. Through various matching algorithms a set of probable solution can be reached to find a student-supervisor match. The match should be as such that maximises both the student's and the supervisor's satisfaction. But unlike the university student and thesis supervisor assignment the post-grad students are not personally acquainted with the supervisors in most of the cases. Also they have various other preferences while searching for a supervisor in this level. So, matching a supervisor with a student during this level is not a straight forward task. It is about taking into account the criteria that influences the students and the criteria that is important to the supervisors, their research interests etc. and then matching them with each other. This is why the traditional matching algorithms is not sufficient in a case like this. Similar problems are addressed in various papers where the criteria of the matching is more than one [10]. We are going to propose a system that works without a clear preference or ranked list while taking account of both students' and supervisors' preferences.

Motivation

Researchers have already addressed this kind of problems and tried to find suitable solutions using matching algorithms. The classical Stable Marriage Problem (SMP) [11] is the central matching problem in this class. An instance of this problem comprises a set of men and women, and each person ranks each member of the opposite sex in strict order of preference. The Student-Project Allocation (SPA) problem - which can be used for other similar problems or Hospitals/Residents problem (HR) [12, 13] is an extension of SMP. In each of the problems in this class, the task is to find a stable matching. Informally, a matching is a set pairs, each of which represents the assignment of an agent from one set to an agent from the other, such that no agent is assigned more agents than its capacity. A matching is stable if no two agents prefer one another to one of their current assignees. If such a pair exist, they could ignore the matching and form a private arrangement outside of it.

A number of related works report implementation of Genetic Algorithm (GA) to find a suitable solution [13–15]. The problem can also be considered as an extension of problems such as the Workers/Firms problem (WF) or Sailor Assignment Problem (SAP) which can also be solved by various Genetic Algorithms [10,16]. In [17,18] the Student-Project Allocation problem is defined explicitly by an objective function and a number of constraints and concludes that as any optimization problem it is possible for constraints to be too tight to permit any feasible solution.

By considering only the criteria that are important for students and supervisors it is only possible to find the matching by generating a ranked list from the given information. As there is a lack of any kind of ranking in the initial data, a suitable ranking from these criteria and constraints for both student and supervisor can be created using Analytic Hierarchy Process(AHP) which provides a method for quantifying the weights of the criteria [19, 20]. Employing this ranking a comparison matrix can be created from which the strength of any match between student-supervisor is calculated using the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) method [21].

Objectives of This Thesis

In this work a solution will be proposed for the student supervisor assignment problem by considering students' and supervisors' preferences. This will be done by surveying students and supervisors to get their preferences.

The objectives of this thesis are as follows:

i. To develop a ranking scheme based on students' and supervisors' preferences using AHP method [19].

ii. To develop a weighted matrix based on the criteria of the ranking using TOPSIS method [21].

iii. To find out a suitable solution for student-supervisor allocation using Kuhn-Munkres algorithm.

iv. To compare the proposed algorithms with Gale-Shapely [11] algorithm for stable marriage and genetic algorithm with informed initialization on large instances.

Our Contributions

Based on our work, our contributions are as follows:

- We collected the data from both students and supervisors about what influences them when looking for a perspective supervisor or student respectively.
- We also collected a numeric value for each of these criteria to determine how the factors' importance vary compared to each other.
- Then these values are fed into the AHP system to find their relative weight.
- Subsequently, a score is calculated for each student by each supervisor and for each supervisor by each student with their individual qualities, preferences and weight matrix.
- From this score the strength of each pair is determined using TOPSIS method.
- Finally, Matching algorithm is applied on this matrix to find suitable matches with highest satisfaction from both sides.

Organization of the Thesis

The rest of the book is organized in the following way.

In Chapter 2, we will show the background and related research studies. How matching algorithms are developed throughout the years and their application in various sectors. Also how the ranking system AHP works and how TOPSIS method calculates the strength of a solution.

After that in Chapter 3, we will discuss about the factors that influences students and supervisor. Then how from the individual priorities can be used to find the weight of the factors and calculate the ranking of the students and supervisors. Then we will discuss the methodology that we use to solve the problem which is formulated.

Later, in chapter 4, we will discuss our implementation strategy to perform the experiments and find the statistics.

In chapter 5, we will have a short conclusion including the future possible research directions.

Chapter 2

Literature Review

In this chapter we are going to present a literature review to our work. First we are going to start with studies done so far regarding matching algorithms and various student-supervisor allocations. Then we will discuss contexts such as AHP and TOPSIS methods that are close to our proposed architecture.

Similar Problems

Generally an Assignment Problem is a problem that basically tries to find suitable matches between two sets of elements given an ordering of preferences for each element. The basic focus of an Assignment Problem is to maximise the satisfaction of all the elements from both sets. But depending on the problem definition an assignment problem can be of various categories. Such As -

Single-Objective Matching Problem A single objective matching is somewhat straight forward where both sets of items have a clear preference list for the other set of items. So the decision can be taken from the predefined preference list. For example - stable marriage problem, networking etc.

Multi-Objective Matching Problem When in a matching problem either one or both set of items don't have a clear preference list and the preference list depends on more than one criterion so that if one item is desirable for an element from the other set based on one criterion, it can be totally undesirable by the same element when another criterion is considered. So a matching algorithm for these kinds of problems can be very complicated. Sometimes these types of problem are first converted to single-objective problem and then solved. Example of this problem are - Sailor-Job assignment, resource allocation etc.

Matching Algorithm

Matching algorithms are originally algorithms used to solve graph matching problems in graph theory. A matching algorithm in graph theory takes a graph G with V vertices as input and outputs Edges E where no two edges share the same two beginning and ending vertices.

Figure 2.1 shows an example of a matched graph -

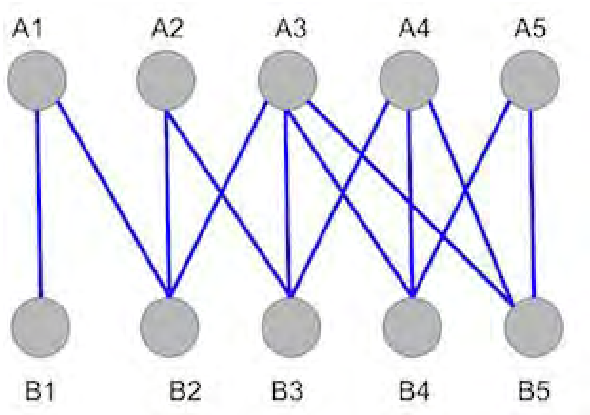

Figure 2.1: Example of a matched Graph

A matching algorithm in graph theory can fulfil different purposes. Some most used matching algorithms in graph theory are Bipartite Matching which has various applications in other sectors of science too.

Bipartite Matching

A Bipartite Graph G is a graph whose vertices can be divided into two independent sets U and V such that every vertices of G either belongs to U or V . A bipartite matching is a set of edges $E(u, v)$ where $u \in U$ and $v \in V$.

Bipartite matching problems with two-sided preferences

Here the participating agents can be partitioned into two disjoint sets, and each member of one set ranks a subset of the members of the other set in order of preference. Example applications include assigning junior doctors to hospitals [8], pupils to schools [22–24] and school-leavers to universities [25–27].

Bipartite matching problems with one-sided preferences

Again the participating agents can be partitioned into two disjoint sets, but this time each member of only one set ranks a subset of the members of the other set in order of preference. Example applications include campus housing allocation [28, 29], assigning reviewers to conference papers [30] etc.

Non-bipartite matching problems with preferences

Here the participating agents form a single homogeneous set, and each agent ranks a subset of the others in order of preference. Example applications include forming pairs of agents for chess tournaments [31], finding kidney exchanges involving incompatible patient–donor pairs [32–34] and creating partnerships in P2P networks [35–38].

Maximum-Weight Bipartite Matching

In a **Maximum-Weight Bipartite Matching** every edge has an individual weight and the goal is to find such a match where the sum of the weights in the matched graph is maximized. There can be more than one maximum matching for a given Bipartite Graph.

Applications of Maximum-Weight Bipartite Matching

There are many real world problems that can be formed as Bipartite Graphs. For example it has applications in greedy algorithms, network flow, shortest path algorithm, job matching, marriage matching, assignment problems etc.

Some Popular Matching Algorithms

Below are some of the well-known matching algorithms.

Stable Marriage Problem

The Stable Marriage Problem(SMP) is widely used in mathematics, economics, and computer science. It finds the stable matching between two sets of same number of elements when an ordering of preferences for each element is given.

A matching is mapping an elements of one set to an element of the other set. A pair $(m_i, w_j) \in E \backslash M$ blocks a matching M, or is a blocking pair for M, if the following conditions are satisfied relative to M:

i. m_i is unassigned or prefers w_j to $M(m_i)$;

ii. w_j is unassigned or prefers m_i to $M(w_j)$.

A matching M is said to be stable if it admits no blocking pair. In other words, a matching is not stable if -

i. There is an element A of the first matched set which prefers some given element B of the second matched set over the element to which A is already matched.

ii. B also prefers A over the element to which B is already matched.

So, a matching is stable when there does not exist any match (A, B) which both prefer each other to their current partner under the matching.

In many practical matching applications where the underlying theoretical model is based on a bipartite matching problem with preferences on both sides, stability is the key criterion to be satisfied. However, when preference lists are incomplete, a stable matching might be smaller (up to 50% smaller in the worst case) than a maximum cardinality matching. In some applications, a limited number of blocking pairs may be tolerated if that enables a larger matching to be found.

But there are some other issues of strategy in stable matching problems too. The question is whether an agent can misrepresent his/her true preferences in order to obtain a better outcome with respect to a given mechanism. Such questions have been the focus of much research by economists traditionally, and an extensive coverage of results up to 1990 appears in Ref. [39]. In the subsequent years, increasingly this line of research has been taken up by computer scientists.

Gale–Shapley Algorithm

The Gale–Shapley algorithm for SMP can be viewed as a centralised matching algorithm. In 1962, David Gale and Lloyd Shapley proved that [11], for any equal number of men and women, it is always possible to solve the SMP and make all marriages stable. It typically starts from a set of matching (which may be empty) and iteratively blocks pairs in order to arrive at a stable matching.

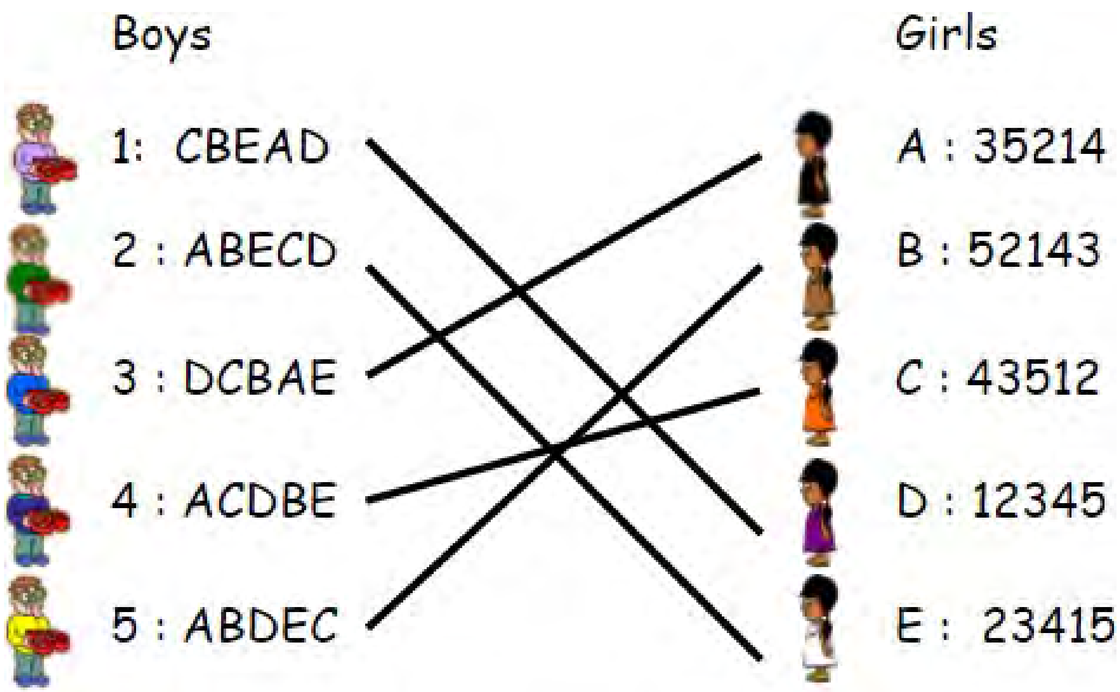

Figure 2.2: Example of Gale–Shapley Algorithm [1]

Given a preference list from all the men and women the Gale-Shapley algorithm is solved using the following steps

i. Each un-engaged man proposes to the woman he prefers most.

ii. Each woman replies tentatively accepts the suitor she most prefers and rejects all other suitors.

iii. In each subsequent round, each un-engaged man proposes to the most-preferred woman to whom he has not yet proposed.

iv. The steps are repeated until everyone in matched.

The Gale-Shapley algorithm guarantees that:

1. Everyone Gets Engaged: For set of equal number of men and women at the end of everyone is matched with one other person.

2. The Marriages Are Stable: Let Adam and Brenda both be engaged, but not to each other. After all the assignments are done, it is not possible for both Adam and Brenda to prefer each other over their current partners. If Adam prefers Brenda more than his current partner, he must have proposed to Adam before he proposed to his current partner. And during this, Brenda accepted his proposal, but in the end has not married him means she must have dumped him for someone else who she likes more, and therefore cannot like Adam more than her current partner. So in the situation where Brenda rejected his proposal, she was already with someone who she likes more than Adam.

The Gale-Shapely algorithm has the following disadvantages:

1. Depending on how it is used, it can find the solution that is optimal either for the participants on one side of the matching, or for the participants on the other side.

2. An individual agent, or some coalition of agents, can falsify their preference list/s (typically by permuting some entries and/or truncating their list) so as to obtain a better partner (with respect to the true preferences) than they would obtain in either the man-optimal or woman-optimal stable matching.

   Ideally there would exist a mechanism for constructing a matching in which it is a dominant strategy for each agent to report his/her true preferences, regardless of whether the other agents are doing likewise. That is, the best outcome for each agent would be obtained by telling the truth, no matter whether the other agents are doing so. Such a mechanism is called a strategy-proof mechanism (also known as a truthful mechanism). Roth [40] showed that, with respect to the man-oriented Gale–Shapley algorithm, it is a dominant strategy for the men to tell the truth. On the other hand he showed that, more generally, there is no mechanism for SMP for which it is a dominant strategy for all agents to be truthful, and hence there is no strategy-proof mechanism for SMP.

To describe strategic results for SMP in more detail, let I be an SMP instance representing the true preferences of the agents, and let M_a (respectively M_z) denote the man-optimal (respectively woman-optimal) stable matching in I. Let C be a coalition of agents who falsify their preferences, and denote by I' the preference lists that result (each agent not in C has the same preference list in I and I').

Dubins and Freedman [41] proved that there is no coalition C of men who could falsify their preferences so as to yield a matching M' that is stable in I' such that every man in C has a better (with respect to I) partner in M' than in M_a. Roth [40] independently proved this for the special case that C comprises a single man. Demange, Gale and Sotomayor [42] extended Dubins and Freedman's result to the case where C can include both men and women as follows. They proved that there is no stable matching M in I such that every member of C prefers (in I) their partner in M to their partner in every stable matching in I. If incomplete lists are permitted, Gale and Sotomayor [43] proved that, as long as two stable matching in I exist, then we can choose C to contain a single woman who could falsify her preferences so as to yield a better (with respect to I) partner than in M_a. They also showed that if C is the set of all women, then each woman can truncate her preference list so as to force the man-oriented Gale–Shapley algorithm to yield M_z in I' (rather than yielding M_a in I). Teo et al. [44] considered the SMP setting and, in particular, the case where there is a single woman w who knows the preferences of all other agents, which are declared truthfully. They showed how to construct, in polynomial time, an optimal cheating strategy for w relative to the man-oriented Gale–Shapley algorithm. However, interestingly, they proved in simulations that it is relatively unlikely that a woman could gain any advantage by cheating. In particular, they generated 1000 random instances of size 8 and found that, for 74% of these, the deceitful woman did not improve from the partner that she would obtain in M_a (the man-optimal stable matching with respect to the true preferences), and on average, only 5.1% of women did improve by cheating. The authors also presented a discussion of school placement in Singapore, that a system based on stable matching would be more appropriate than the algorithm that was in place at the time of writing.

3. It has the time complexity $O(n^2)$.

The Hospitals/Residents Problem

The National Resident Matching Program (NRMP), a United States-based private non-profit non-governmental organization created in 1952, the Canadian Resident Matching Service [45] and the Scottish Foundation Allocation Scheme [46] handle the assignment of graduating medical students to junior doctor appointments in hospitals in their respective countries. At the heart of these matching schemes are efficient algorithms that essentially solve some variant of the Hospitals/Residents problem (HR). In large-scale matching schemes of this kind, many participants, specially the large popular hospitals, may not be able to provide a genuine strict preference order because they may have a very large number of applicants, and may be indifferent about their preference lists. The most general form of indifference can be modelled by the Hospitals/Residents problem with Partially-ordered lists (HRP). Programs and applicants each provide a ”ranked list” to the NRMP. Programs list applicants, ranked in order from most to least preferred, whom they wish to train. Similarly, applicants rank programs according to their personal choice. For those applicants who wants to be considered as a couple, the rank order lists include pairs of program choices that are considered simultaneously by the matching algorithm. Perhaps the best-known example is the NRMP, which handled over 38,000 applicants in 2012. Perhaps the largest existing centralised matching scheme is the one that handles higher education admission in China, were around 10 million applicants to Chinese higher education institutions in 2007 [47].

The Assignment Problem

The assignment problem is a fundamental combinatorial optimization problem. In a weighted bipartite graph it finds a matching of a given size, in which the sum of weights of the edges is a minimum or a maximum based on the problem definition. A common variant of this problem is finding a maximum-weight matching. Where the goal is to find all the matching that maximises total weight, for a bipartite graph.

Kuhn–Munkres Algorithm

The Hungarian matching algorithm, also called the Kuhn-Munkres algorithm, is an algorithm that can be used to find maximum-weight matching in assignment problems. It was developed and published in 1955 by Harold Kuhn [48, 49]. Later James Munkres reviewed the algorithm in 1957 [50]. A bipartite graph can be represented by an adjacency matrix, where the weights of edges are the entries.

For example if there are three workers A, B and C who can do three tasks X, Y, Z at different cost as given in the table below:

	X	Y	Z
A	2 dollars	3 dollars	3 dollars
B	3 dollars	2 dollars	3 dollars
C	3 dollars	3 dollars	2 dollars

The target is to find the best job-worker match so that all the three jobs can be done with minimum cost. The problem can easily be represented into a $n \times n$ matrix where the element in the i-th row and j-th column represents the cost of assigning the j-th job to the i-th worker. When Kuhn–Munkres Algorithm is applied to the above table, it would return the minimum cost, which 6 dollars, achieved by having A doing X, B doing Y and C doing Z.

Student-Project Assignment

In this section we consider the problem of allocating students to projects based on their preference lists and capacity constraints of the projects. This problem is known as Student–Project Allocation problem (SPA). The Student-Project Assignment problem is a bit different from other assignment problems. As in other assignments one element from set A is only matched with one other element of set B. But is case of Student-Project Assignment a project/supervisor can be matched with one or more students.

During the upper level of their degree programs, students are often required to undertake project in a field of their choice as a part of the said program. Typically more projects are offered where usually the total number of project places are larger than the number of students to provide the students a choice. Also, typically each supervisor will offer a variety of projects, but does not necessarily expect that all will be taken up.

Each student has preferences over the available projects that he/she finds acceptable, whilst a supervisor will normally have some form of preference list over the projects he/she offers and/or the students who find them acceptable. There is usually a limit on the number of students that can be assigned to a particular project, and the number of students that a particular supervisor can supervise. .

Variants of SPA arise according to the nature of the preference lists that supervisors provide. In the case of some centralised matching schemes that assign students to projects, supervisor preferences are not permitted [17, 51, 52]. However, some universities, for example, the Department of Computer Science at the University of York permits supervisor preferences over students in its centralised student–project allocation process [53–55]. This leads to the first variant of SPA, namely the Student–Project Allocation problem with supervisor preferences over Students (SPA-S) [12], in which each supervisor *S* ranks in order of preference the students who find acceptable at least one project that *S* offers. Such a preference list may reflect *S*'s assessment of the students' academic suitability for her projects.

An alternative variant of SPA is the Student–Project Allocation problem with supervisor preferences over Projects (SPA-P) [56, 57], in which each supervisor *S* ranks in order of preference the projects that he/she offers. This preference list may reflect the possibility that *S* prefers to supervise projects that are closely connected with his/her research, whilst the remaining projects that *S* offers (perhaps only proposed to ensure that the students have adequate choice) have a lesser priority.

The final variant is a hybrid version of SPA-S and SPA-P. In the Student–Project Allocation problem with supervisor preferences over Student–Project pairs (SPA-(S,P)), each supervisor *S* has a preference list that depends on not just the students who find acceptable a project that *S* offers, but also the particular projects of *S*'s that these students would undertake.

Anwar et al. [17] were one of the pioneering authors in providing a computational solution to the student-project allocation problem. The article introduces two different integer programming models: one to allocate students to projects while minimizing the projects supervised by supervisors, and another to maximize the students' satisfaction according to their preferences on group projects they wish to be allocated in. In this setting, supervisors propose a list of projects and students provide a rank of four projects they want to be allocated in. Both integer programming models were tested on a real data-set consisting of 60 projects, 22 staff members, and 39 students. Similarly, [58] introduces the use of genetic algorithms for solving the student-project allocation problem where the students provide a ranked list with their most preferred projects, and each student is allocated to a project from the provided list, where the students have to carry out the projects individually. The algorithm was tested with real data consisting of 25 students and 34 projects, and also with problems created from data provided from the OR-library [59]. These models only take into consideration the preferences of the students, but do not consider the supervisors' preferences with regards to projects and students and the workload of the supervisors. In addition to this, they can only optimize a single objective function which does not allow decision makers to trade-off between the students' and the supervisors' preferences. Abraham et al. [12] focused on solving the student-project allocation problem from an optimal perspective. The authors assume that a the supervisors provide a list of projects. The students provide a ranked list of their most preferred projects, while supervisors explicitly rank those students who want to be allocated in his/her projects. Under these assumptions, the authors provide two linear algorithms to find stable matching: one from the perspective of the students' preferences, and another from the perspective of the supervisors' preferences. While by employing these algorithms it is guaranteed that a solution will be found, these solutions will wither be optimal solution for the students or the optimal solution for the supervisors, with no trade-off opportunity provided to the decision makers. In addition, these algorithms do not take the workload of the supervisors into consideration, so they may produce unbalanced solutions. Finally, it should also be considered that supervisors may not know the students well enough to rank them explicitly, or it may be unfair for students with lower marks as they will most likely end up in the last rank positions in supervisors' preferences.

Later on, Manlove & O'Malley [57] study the student-project allocation problem in a scenario where students and supervisors have preferences over a set of projects. Both projects and supervisors have capacity constraints. Under these conditions, the authors prove that stable matching can have different cardinalities, and thus the objective is that of finding the stable matching with a maximum cardinality. Solving this problem is NP hard, but the authors provide a student oriented approximation algorithm with a performance guarantee of 2 (i.e., only guaranteeing half of the cardinality of the maximum stable matching) and polynomial complexity. Iwama et al. [56] further narrowed down this bound to a range between 1.5 and 1.10. The proposed algorithms focus on optimizing the students' preferences, and does not consider the supervisors' preferences, their workload or the lower quota constraints. Salami and Mamman propose another genetic algorithm for scenarios where students have complete preferences on supervisors, and supervisors have a maximum supervision quota [13]. However, this algorithms also do not consider the supervisors' preferences or their workload balance. Victor Sanchez-Anguix et al. in their paper [60], proposed a solution to find suitable assignment between students and supervisors from their preference list where the supervisors has a lower and upper boundary for how many students they can take. Their approach is based on students' and supervisors' preferences on project topics rather than projects. This gives an advantage as it does not require the supervisors to propose projects prior to the allocation and they can be negotiated with students according to their research interests. Furthermore, it does not discriminate students according to their previous results as supervisor preferences' are based on topics rather than students. They considered both the students' and the supervisor' preferences by adopting a multi-objective approach that provides decision makers with flexibility to trade-off between objectives as it estimates Pareto optimal solutions.

Although the SPA problem model and its variants are introduced and motivated in the context of Student–Project Allocation, they are equally valid in other scenarios, for example where applicants apply for posts at large organisations, which are split into several departments.

Multi-objective Matching Problems

Multi-objective optimization is an area of decision making where there are more than one criteria and is concerned with mathematical optimization problems concerning simultaneously optimizing more than one objective function. Many of the matching problems in science, economics, and finance require decision making depending on more than one criteria. So a preference list for both sets depends on various factors. Optimizing such a matching is a bit different as there is no direct preference list. Various Authors has proposed different algorithms to find a suitable algorithm for such problems. Deb et al. [61] proposed NSGA-II, a well-known GA schema for estimating pareto optimal solutions in multi-objective problems. Various problems in academic and industrial areas have several conflicting objectives that need to be optimized simultaneously [62], they are called multi-objective optimization problems (MOPs). The most commonly adopted notion of optimum in multi-objective optimization is Pareto optimality, which refers to finding the best possible trade-offs among the objectives of a multi-objective problem. These solutions constitute the Pareto optimal set. And the image of this Pareto optimal set is called the Pareto front. Among the different techniques available to solve MOPs, multi-objective evolutionary algorithms (MOEAs) is the most popular one mainly because of their flexibility and ease of use. Modern MOEAs normally aim at producing, in a single run, several different solutions, which are as close as possible to the true Pareto front [62]. For several years, MOEAs adopted a selection mechanism based on Pareto optimality. However, in recent years, it was found that Pareto-based MOEAs cannot properly differentiate individuals when dealing with problems having four or more objectives (many-objective optimization problems [63]). This motivated the researcher to develop other alternative selection schemes among which performance indicators has been the most popular choice until now. In indicator-based selection the primary target is to identify the solutions that contribute the most to the improvement of the performance indicator adopted in the selection mechanism.

Decision Making and Ranking

We are all fundamentally decision makers. Most of our conscious or unconscious activities are the result of some decision. The information we gather about something help us to understand the situation better, in order to develop good judgements we use these knowledge to make decisions about these occurrences. But what we gather is not always information or useful for improving our understanding and judgements. If we only make decisions intuitively, we are inclined to believe that all the information we gathered are useful and the larger the quantity, the better. But that is not always the case.

There are many examples, which show that too much information can sometimes be as bad as little information. Knowing more does not guarantee a better grasp on the issue. It is illustrated by some author's writing, "Expert after expert missed the revolutionary significance of what Darwin had collected. Darwin, who knew less, somehow understood more" [64].

To make a decision we need to know the problem, purpose of the decision, the criteria that influence the decision, their sub-criteria, stakeholders and the groups that will be affected by the decision. Also we need to have a clear idea about the alternative actions that can be taken. Among them we then try to determine the best alternative, or in case of resource allocation, we need to put priorities on the alternatives to allocate the resources.

Decision making, for which we gather most of our information, has become a mathematical science today [65]. It helps us to formalises the thinking process so that it becomes transparent what we have to do to make better decisions from all aspects. Decision making involves many factors and sub-factors that influences that decision and they are used to rank the alternatives of the decision. One need to create priorities among those factors so they can weigh those alternatives and make a good decision. Those factors may be intangible, and may have no concrete measurements that will help to rank the alternatives, and creating priorities for the factors themselves in order to weigh the alternatives can be a very challenging task.

The measurement of intangible factors, as they can not be described using numbers is the hardest part to formalise using mathematical science. So far, mathematics has assumed that all things can be assigned numbers from minus infinity to plus infinity in some way and all mathematical modelling generally are based on this assumption.

Naturally, all this is predicated on the assumption that one has the essential factors and all these factors are measurable. But there are many more important factors that we do not know how to measure than there are ones that we have measurements for. Knowing how to measure such factors could conceivably lead to new and important theories that rely on many more factors for their explanations. If we knew how to measure intangibles, much wider room would be open to interpret everything in terms of many more factors than we have been able to do so far scientifically. One thing is clear, numerical measurement must be interpreted for meaning and usefulness according to its priority to serve our values in a particular decision. It does not have the same priority for all problems. Its importance is relative. Therefore, we need to learn about how to derive relative priorities in decision making.

It is often the case that ordinal preferences naturally arise from human subjects who are participants of a centralised matching scheme and able to arrive at a notion of first, second, third choice, etc. However there are some interesting cases where quantitative, objective data do in fact give rise to ordinal preferences, such as the following:

i. Junior doctor allocation in Scotland (applicant “scores” based on academic performance and assessment of application forms give rise to ordinal preferences for the hospitals) [46, 66].

ii. School choice in New York and Boston (children have “priorities” based on factors such as whether they are in the walk zone and whether they have siblings at the school already, and these priorities translate into ordinal preferences for the schools) [22–24].

iii. Higher education admission in Hungary (again, the academic performance of the applicants gives rise to ordinal preferences on the part of the universities) [67, 68].

iv. P2P networking (measures of download / upload bandwidth, latency and storage capacity give rise to ordinal preferences of nodes in a communication network over their peers) [35–38, 69].

Thomas L. Saaty [19] in the 1970s developed a strong and helpful method for managing qualitative and quantitative multi-criteria elements involving in decision-making behavior. This model is called Analytical Hierarchy Process (AHP) and is based on a hierarchical structure.

Analytical Hierarchy Process (AHP)

The Analytical Hierarchy Process (AHP) is a structured technique for organizing and analyzing complex decisions, based on mathematics and psychology, where numeric values like price, weight, area, and even non-numeric factors such as feelings or satisfaction, can be converted into measurable numeric relations. The core idea of AHP is that it compares all the factors in pairs instead of sorting or voting or the free assignment of priorities.

AHP is used over the years in various sectors such as science, business, industry, healthcare and education.

The strength of AHP is that it is flexible enough to be integrated with different techniques like Linear Programming, Quality Function Deployment, Fuzzy Logic, etc. This allows it to be used in various combined methods, and hence, achieve the desired goal in a better way. Lai et al. [70] used AHP for software selection called Multi-media Authorizing System (MAS). They used the group decision-making technique which included six software engineers and evaluated three products of MAS. They used a four-level hierarchy of the pair-wise comparison. The criteria that were used are development interface, graphics support, multi-media support, data file support, cost effectiveness, and vendor support which were evaluated in level three. Six software engineers were trained about the use of AHP, and then asked to pair-wise compare these different criteria. Expert Choice software was used to felicitate ease in computation. To decide the selection consensus, geometric mean methodology was chosen and the production software with a large geometric mean value was selected.

Al Harbi [71] applied AHP in the field of project management to select the best contractor. He developed a hierarchical structure for the pre-qualification criteria and the contractors who wish to qualify for the project. A total of five contractors were considered in the case study. They were evaluated based on the criteria of experience, financial stability, quality performance, manpower resources, equipment resources and current workload. The contractors were compared with each-other pair-wisely for the criteria mentioned above. Ranking among all those criteria was also done to find out the overall priority of each contractor. Based on this resultant priority, the best contractor was selected who had a highest overall priority value.

A four-step algorithm for locating and selecting the convenience store (CVS) is presented by Kuo et al. [72]. They used AHP as it certainly has advantage over the conventional methods. Jung and Choi [73] presented optimization models for selecting best software product among the alternatives of each module in the development of modular software system.

Forgionne and Kohli [74] used AHP to evaluate the quality of journals, with a methodology for consolidating the multiple-criteria into an integrated measure of journal quality, with discussion on data collection process. Badri [75] used AHP as an aid in making location allocation decisions. He claimed that the methodology, in the volatile complex decision environment, could help the facility planning personnel to formulate the location strategies. Lee and Kwak [76] presented a case study to plan the information resource in a health care system. The objective of the planning was to design and evaluate a model that would effectively help in the planning of the health care system. The model proposed by the authors, incorporated goal programming to reflect the multiple conflicting goals, and to provide a solution to the multidimensional allocation planning. AHP plays a crucial role in decomposing and prioritizing these different goals and criteria in the planning scenario.

Padma and Balasubramanie [77] used AHP to develop a decision aid system in order to rank risk factors associated with the occurrence of musculoskeletal problems in the shoulder and neck.

AHP has been applied to take decisions in education sector as well. Miyaji et al. [78] solved the education decision problem of examination composition using AHP. The test results and the selection of questions are utilized for the purpose. The authors argue that the results of the examination are used to get an idea about the students degree of understanding, and to help them to learn individually. It is a critical work to choose questions for the examinations from a huge database. The question selection becomes more complicated if content form, correct answer rate, distribution of difficulty degree, size, etc. are to be considered. To overcome this problem they proposed a two-stage decision support system. Firstly, some plans are presented using branch and bound methods. The teacher then decides on the plan. Two different factors are considered for framing of the different alternatives: 1)whether a student can give an answer within the range of examination content, and 2) whether the students can solve the problem in the given time frame. A hierarchical structure of AHP is formulated for the necessary composition and selection of the examination problem.

Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS)

TOPSIS is a simple ranking method in conception and application which is developed by Hwang and Yoon in 1981. The standard TOPSIS method attempts to choose alternatives that simultaneously have the shortest distance from the positive ideal solution and the farthest distance from the negative ideal solution. The positive ideal solution maximizes the benefit criteria and minimizes the cost criteria, whereas the negative ideal solution maximizes the cost criteria and minimizes the benefit criteria. TOPSIS makes full use of attribute information, provides a cardinal ranking of alternatives, and does not require attribute preferences to be independent [79, 80]. To apply this technique, attribute values must be numeric, monotonically increasing or decreasing, and have commensurable units. This wide range of real-world applications for the TOPSIS method imposed a strong motivation for categorizing applications across different fields and specific sub-areas such as resource management, Engineering and Manufacturing Systems, Business and Marketing Management etc.

Aydogan [81] proposed integrating AHP and fuzzy TOPSIS to evaluate the performance of four aviation firms using five important dimensions: performance risk, quality, effectiveness, efficiency, and occupational satisfaction.

Kelemenis et al. [82] proposed a multi-criteria approach based on fuzzy TOPSIS group decision-making to select a middle-level manager in a large IT Greek firm. Boran et al. [83] employed an intuition based fuzzy TOPSIS approach to select appropriate personnel from candidates when selecting a sales manager at a manufacturing company.

Mahmoodzadeh et al. [84] using fuzzy AHP and TOPSIS technique proposed a new method for project selection problem. After reviewing four common methods of comparing alternatives investment (net present value, rate of return, benefit cost analysis and payback period) they use them as criteria in AHP tree. In this methodology by utilizing improved Analytical Hierarchy Process by Fuzzy set theory, they calculate the weight of each criterion. Then by implementing TOPSIS algorithm, assessment of projects has been done. Hota et al. used Fuzzy TOPSIS method to rank teachers in higher education. The took 10 alternatives (Teachers) as T1, T2,. . . , and T10 and five criteria: Knowledge, Communication, Explanation, Method of Teaching, and Experience as C1, C2, C3, C4, and C5, respectively. The Teacher's nature and behaviors with respect to each criterion are calculated with the help of methodological steps, then closeness coefficient of each teacher is calculated by using Fuzzy technique and the weights are determined according to linguistic terms (Bad, Average, Good). Distance of each alternative from fuzzy positive ideal solution (FPIS) and fuzzy negative ideal solution (FNIS) with respect to each criterion are calculated using FTOPSIS steps then the results of all alternatives distance from FPIS and FNIS are obtained as Sj+ and Sj - for 10 alternatives. Using these values closeness coefficients of 10 alternatives are calculated according to the closeness coefficient of 10 alternatives and finally the ranking order of 10 alternatives are determined.

Furthermore, in recent years, TOPSIS has been successfully applied to the areas of human resources management [85], transportation [86], product design [87], manufacturing [88], water management [89], quality control [90], and location analysis [91]. In addition, the concept of TOPSIS has also been connected to multi-objective decision making [92] and group decision making [93]. The high flexibility of this concept is able to accommodate further extension to make better choices in various situations. This is the motivation of our study.

Conclusion

Though over the years, many systems are developed to solve the assignment problem, most of them basically focuses on a known and limited set of parameters. From the stable marriage algorithm to its various derivative versions, the main focus was to assign partnership with a straight preference list. But when no clear preference list is present, in a situation like that we can create a preference list from the basic data collected from the students and supervisors. In this way the problem can be reduced to a classical assignment problem.

On the other hand, the AHP and TOPSIS methods are used for ranking in various problems especially for pairwise comparisons. So using it in the student-supervisor assignment problem will lead to the possibilities of using these methods even in problems where direct pairwise comparison is complex.

Chapter 3

Proposed System

In this chapter we are going to explain the ranking technique we applied based on different criteria both from students' and supervisors' perspectives. While searching for a grad-school/supervisor a student has to take many things into considerations. If the school is located in a friendly environment, what is its research reputation, how experienced is the supervisor, what is his/her research area, how many publications he/she has and so on. Similarly a supervisor has his/her own preferences while accepting grad-students. How was their previous academic record, whether they're involved in research etc.? If the student has to go through the possible pool of grad-schools/supervisors manually checking all his/her criteria and weighing which option is better than the other it will be a time consuming work. Similarly for a supervisor doing the checking manually will be an enormous task too. To find a way around this problem, we propose a ranking system customized for each student/supervisor. This system will first recognize the weightage of those criteria the student/supervisor looks for, and then it will rank the available supervisors/students who fall into those criteria better in a descending order. In this way, looking at the rank will tell a student/supervisor who will match best for his/her choice. Finally using these rankings we can also find all those matches that maximize the satisfaction of both sides in a fixed situation.

Later, we will show how this system fares according to the satisfaction of the matches among the two parties.

System Definition

Each student generally has some individual criteria that he/she focuses on more while choosing from a pool of supervisors. So a rank based on a specific students' demand can be very different from the next student who may be focused on different aspects of a supervisor and create a rank based on his/her criteria. For one student his/her list of ranked supervisors can be very different from the next student. The same is also true for each supervisor too.

Ranking System

A ranking system is a good way to systematically categorise a group of items according to some criteria that is applicable to them. The order of the items generally shows their comparative ability to do a certain job. An item 'ranked higher' in one list means it can do that certain job better than the items below it. A ranking, in no way, is a fixed property of that item. The same group of items can be ranked differently for different categories. It can be of different order for other contexts, or in a different system and so on.

So while ranking the supervisors from students' point of view we have to consider the factors and how they influence students while choosing a supervisor and vice versa. Ultimately we created an individual ranking system for each student and supervisor depending on their personal priorities. When we have a fixed number of students and supervisors from these two way rankings a matching can be found that satisfies the highest number of students and supervisors.

Major Factors while Ranking the Students

While a supervisor accepts students for his/her undergraduate/postgraduate supervision he/she generally focuses on a few properties of the student. A supervisor himself has to play a significant role in the successful completion of a postgrad program. Brown and Atkins [94], in their study, suggested that effective supervisors must be competent researchers, must be able to reflect this competence in research practices, and must be able to analyze the knowledge, techniques, and methods that make their supervision effective. Pearson and Kayrooz [95] categorized the tasks and responsibilities of supervision into four groups: those related to the progress of the candidate, mentoring, coaching in the research topic, research methodology, and how to write the dissertation, and sponsorship of the student's participation in academic or professional practice. In their research [96] the authors tried to find out the roles and abilities of the supervisors from the perspective of the supervisors themselves. All the supervisors they interviewed see their main role as teachers of research topics; only some consider that they are also training their students in research, and only few of these believe they are conducting research with their students. So, while taking a student a supervisor has to be confident that he/she will be able to satisfy these roles with that specific student. To achieve these they generally focus on students' previous result or their experience in research to get a probable idea about the students' ability. While ranking the students we focused on three major criteria a supervisor considers while taking in a student. The result or grade acquired by the student during his/her course of studies, the student's area of interest or expertise for research, and his /her previous experience in research or publications. The figure 3.1 shows the result of the survey done during the undergraduate thesis by Humayara Binte Rashid and Annita Tahsin Priyoti [97] on the thesis supervisors of CSE department of BUET. They are asked to rank the four topics according to their preference from 1 to 4 when choosing a student for supervision. From the figure it is clear that supervisors ranked field of interest, publication, previous research work, and programming related work as 1st, 2nd, 3rd and 4th priority.

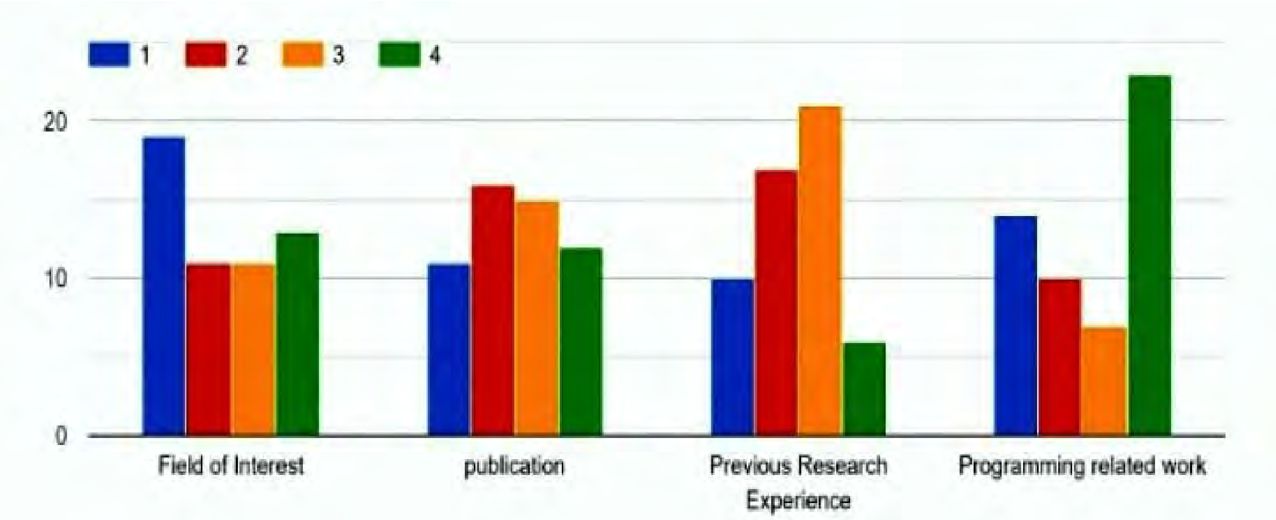

Figure 3.1: factors influence a supervisor while selecting a student

Next we'll discuss about these three criteria that influences a supervisor shown in Figure

3.1.

Explanation of selecting the Criteria that Influences Rankingof the Students

For ranking the students we've decided to use the first three more general criteria that are important to the supervisors.

Previous Result of the Students

Though not a widely popular method among the students or the supervisors, our known education system right now are yet to find a better system that summarizes and examines a students' ability to learn. So we're still largely dependent on examinations and use the results to judge or determine a students' quality. Grade Point Average (GPA) is one of the most widely used measures of undergrad/graduate study success [98]. Most educational institutions these days use it to place a student in a certain range on the marking scale. A grade point is a numeric or alphabetical symbol that a student is awarded for a range of marks. A student over the course of a specific degree or stage of education accumulates various grade points in different examinations and at the end of the degree all the grade points are used to calculate the final GPA. Though critics of the practice may argue that averaging grades over a semester, year, or school tenure can misrepresent student learning, particularly learning growth over time, and that it can adversely affect a student's academic performance, educational confidence, and sense of self-worth. But it still serves a way to estimate a students learning ability and standing as a student. Although Standardized tests such as the Graduate Record Examination (GRE) are a reasonable alternative to the frequent use of undergraduate grade point average (UGPA) in selecting graduate students [99] as test scores resulting from the three GRE parts measuring Verbal, Quantitative, and Analytical abilities can be easily compared among large numbers of applicants with different educational and national backgrounds it is not yet as widely practised as GPA.

Area of Research Interest

A research area is a specific topic or subject area that a student wants to focus on during his/her research or an area in which the problem definition falls. A research area can be vast and a student can focus on some of its topics. But choosing a research area is vital because it reflects what interests him/her the most. Depending on the graduate program and/or supervisor, students may need substantial disciplinary knowledge and academic skills to be prepared to pursue the degree. Prior knowledge may be particularly important in science, engineering, and mathematics related disciplines [100]. Research across different field supports that graduate students need a robust skill set in several key areas to successfully complete the degree. A student's field of expertise or a current trending topic or available projects influences him/her the most while choosing the area. So when a supervisor accepts a student he/she is most concern with the students area of interest as it'll be easier to motivate a student if he/she feels a close connection with the research topic. A supervisor who has years of research experience as both a student and later as a supervisor generally would like to work with someone who has already some ideas about the topic he/she was going to work with rather than with a student, whom, the supervisor have to build up from the beginning. That's why a supervisor is vastly influenced by the intended student's area of interest or the area he/she has already worked with in their previous educational life.

Published Work

Usually a research work is always done with the intention of publishing it to the grater community. A work without the intention of being a way to help people or enhance the knowledge of people is naturally useless even though every research done in every university or by every student is not always publishable due to lack of quality or standard, uniqueness or necessity etc. So, a student's ability can be measured through the number of his published works too. A student who already has some idea about quality research work is always going to be an extra hand and resource for a supervisor. Keeping this in mind some supervisors insists on taking in students who already have publication(s) in international or national or regionally reputed journals. And even it is hard to always come by, a student who has published his/her work before, it is always going to earn him extra points if they has one or more. Thus number of publication(s) is also a factor that influence the supervisors greatly too. That is why we consider these three factors while ranking the students from the supervisor's perspective.

Major Factors while Ranking the Supervisors

While a supervisor's task for finding a suitable student is hard it is a near impossible task for the students because a student's future plans, career etc. hugely depend on his/her successfully completing a degree program. So in the initial stage finding a supervisor can cause various mental stresses for the students. That's why a student has to carefully search for a suitable program and subsequently the supervisor. As was the factors that were important for the supervisors some of those factors also are important for the students too. But there are also some additional factors that influence the students. Over the years, various researchers have conducted numerous studies to identify the factors affecting the students in choosing higher education institutions and supervisors. Factors such as location, cost, academic programs, advertising efforts, career prospects, parents' influences etc. were deemed to be capable in influencing people's choice of higher education institutions [101–104]. Some additional factor are found by the studies [9, 105, 106] who suggest that graduate student decisions are affected by the academic reputation of the institution, program quality and size, price/cost, financial aid, geographic location, contact with faculty, and a student's individual characteristics such as academic ability and achievement. Kallio [107] found some additional practical and reputational considerations to be relevant too, including spouses' plans, the ability to work in one's current job, cost beyond financial aid, program quality, research opportunities, campus life, social opportunities, and recruitment efforts by the faculty of the program. Poock and Love [108] affirmed the factors identified by [9] and [107] as important for doctoral students in higher education programs. So among these factor we can find that while searching for a suitable post graduate program or a supervisor a student generally focuses on a lot of factors among which we put emphasis on the four factors that are most related to the supervisors - the country or region, research area, experience level of the supervisor, published works etc. The figure 3.2 shows the result of the survey done during the same thesis [97] on the thesis students of CSE department of BUET. They are asked to rank four topics according to their preference from 1 to 4 when deciding to approach a supervisor. From the figure it is clear that for majority of the students programming related work got the majority of the vote and ranked as 1st and teaching experience, publication and field of interest was ranked as follows 2nd, 3rd and 4th.

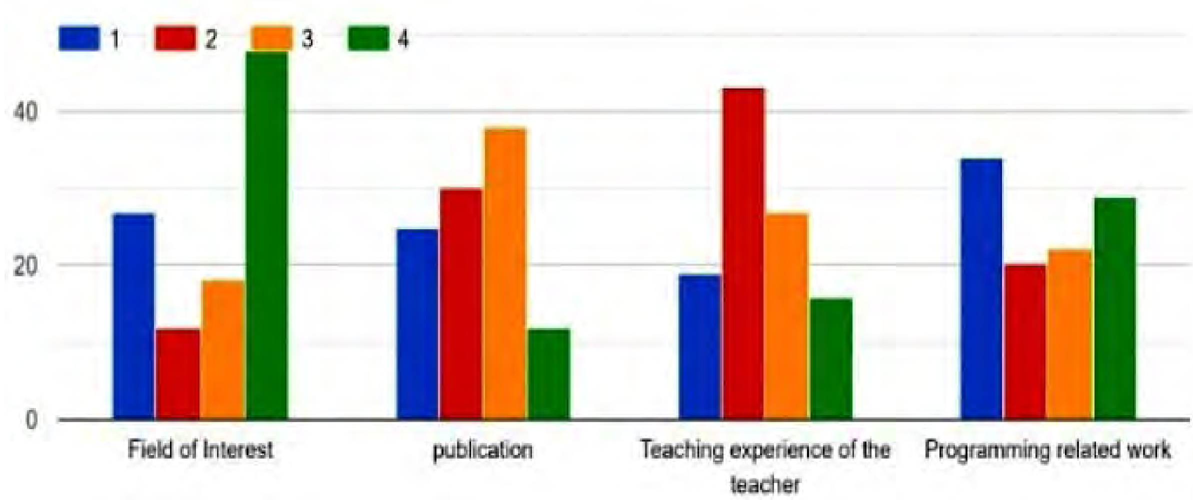

Figure 3.2: factors influence a student while selecting a supervisor

On the other hand during Next we'll discuss about these four criteria that influences a students.

Area of Research Interest

It is important for a student to choose research area that is interesting to him/her professionally, as well as, personally. Experienced researchers note that "a topic in which you are only vaguely interested at the start is likely to become a topic in which you have no interest and with which you will fail to produce your best work" [109]. Ideally, a student's research area should relate to his/her future career path and potentially contribute to the achievement of their career objectives. So to find a suitable research area a student generally focuses on the following steps.

Keeping up with Current Researches A research can only be fruitful if it adds something to the current understanding of things in the world. There is no point in researching about something that is not relevant or useful in the current times. That's why one can get ideas about different research areas by reading recent published works, attending seminars or getting acquainted with various works done by scientists and potential supervisors.

Discussing the idea with others When looking for a research area it is important for a student to find something that interests them. Working on an interesting question helps them in making significant discoveries. Discussion with someone who has previous experience about research and the pros/cons in their research field can help the student to decide on a area or topic he/she will be comfortable with.

Defining a concrete question A research area can be very vast and it might be impossible to cover it in a short amount of time. Defining a specific question about the problem the student wants to study is very important. A primary hypothesis can be then modified and redefined as he studies it intensely and experiments.

So we can see that finding a favourable research topic can be done in various ways and it is a very important start for finding a suitable program/supervisor. That's why it plays a significant role in ranking a supervisor from the point of view of the student. A supervisor with similar area of interest is a much more desirable match for any student than someone with a different area of interest.

Country of the program/supervisor

When choosing a program/supervisor a student not only chooses a definite future for him/her but also sometimes chooses a certain regional destination for him/her and their families as well. The country of the program/supervisor can affect a student in many ways. Such as -

Future Jobs and Welcoming Environment As in the current world the best way of relocating to another region/country and finding a suitable job is to have a degree from a reputed institution of that region/country, more and more students these days primarily focuses on the region/country more than anything while deciding about a program. A country which is more welcoming to foreigners, has more job opportunities, brighter future for the family is always a desirable destination than others.

Funding Opportunities Education is costly in most cases and for a student it is always nearly an impossible task to bear the total cost of a degree from his/her pocket. That's why students are always more eager to consider a program/supervisor which offers funding for the duration of the study.

Geographical Location and Weather As peoples' ability to adapt in a new place and environment sometimes largely depends on the geographical position and the weather of that place, it is also an important factor that plays a role in choosing a region/country.

Published works of the Supervisor

A work published in a reputed journal is always an achievement for a student and sometimes a specific requirement for the degree too. It can help the student by boosting their current career and create new career opportunity. A supervisor with more published work thus becomes more desirable for any student with the view of publishing their work. The student is generally right to assume that a supervisor with more publications in likely to encourage their students more to publish their works too.

Experience of the Supervisor

A supervisor's experience is also a factor that greatly entices a student while considering a suitable supervisor. Generally an experienced supervisor is more likely to have suitable projects and funding options under their belts. So while ranking a supervisor a student put much importance on the years of experience of the supervisor too.

Problem Formulation

Given a set of supervisors and a set of students, where they are to be matched with each other and both the student and supervisors does not have any clear preference list of the other, rather each student has some preferred criteria that they found important while searching for a supervisor. And each supervisor has some preferred quality they considers while accepting a student's application. Now the problem is to find a personalized ranking for each supervisor and student. Those supervisors who match the criteria preferred by a student will be ranked higher in the student's preference list. Same is true for the supervisor too. So looking at the list, it will be clear that the persons ranked higher are a more desirable match than a person who is ranked lower.

Finding Suitable Match

After making a ranked list for the students and supervisors the next challenge is to find a suitable matching that satisfies both the supervisors and students. If a student is matched with a supervisor ranked high in his/her list then that will be a satisfactory match for the student. On the other hand, a supervisor matched with a student who fulfils his/her criteria more will be desirable for him/her. But to set the satisfaction criteria in such a way that ensures most satisfaction on both parts in all matches in the real challenge.

Proposed Solution

In this section we are going to discuss how we are going to use these factors to create a ranking system for both students and supervisors and then use this ranked list to find out the suitable matching between them.

Ranking the Students

The students are ranked using the following steps:

A. Finding Weight of the Factors for Supervisors

As we have said before, we have considered three factors while ranking a student from a supervisor's perspective - CGPA, area of interest and number of publications. But not all supervisors will consider these factors with same importance and also will not follow any fixed sequence while sorting them in ascending or descending order of their importance. So, we needed a weight system that will put definite numeric values for each of these factors on a scale according to each supervisor.

The steps followed to find personalized approximate weight of the factors according to each supervisor:

i. Each supervisor assigns a point (between 1 and 9) for each of the factors/criteria. Higher the point is the criterion has more priority.

ii. A 3×3 matrix M is created using these values such as

$$M(i,j) = \frac{point \quad given \quad to \quad factor \quad i}{point \quad given \quad to \quad factor \quad j} \tag{3.1}$$

For example, if for a supervisor, students' numbers of publication's point is 5 and for area of interest it is 9 then the matrix will look like

factors	*area of interest*	*publication*	*CGPA*
area of interest	1	$\frac{9}{5}$	...
publication	$\frac{5}{9}$	1	...
CGPA	...	...	1

iii. AHP method is applied on this matrix using the steps described in section 3.3.6.

iv. The final matrix is a 3×1 matrix where each value represents the relative weight between 0 and 1 of the respective factor.

Algorithm 1 shows the pseudocode for finding the weight matrix for an individual supervisor.

Algorithm 1 Finding Weight Matrix for a Supervisor

Input: FS = FS_1, FS_2, FS_3 = A set of 3 points given to 3 factors by a supervisor

Output: Weight Matrix W = W_1, W_2, W_3

Set i ← 0; j ← 0;

for i = 0 to 3 **do**
    for j = 0 to 3 **do**
        $comparisonmatrix_{ij} = FS_i / FS_j$

Set T1 ← $comparisonmatrix_{00} + comparisonmatrix_{10} + comparisonmatrix_{20}$
Set T2 ← $comparisonmatrix_{01} + comparisonmatrix_{11} + comparisonmatrix_{21}$
Set T3 ← $comparisonmatrix_{02} + comparisonmatrix_{12} + comparisonmatrix_{22}$

for i = 0 to 3 **do**
    for j = 0 to 3 **do**
        if j = 0 **then**
            $comparisonmatrix_{ij} = comparisonmatrix_{ij} / T1$
        else if j = 1 **then**
            $comparisonmatrix_{ij} = comparisonmatrix_{ij} / T2$
        else
            $comparisonmatrix_{ij} = comparisonmatrix_{ij} / T3$

sum ← 0;
for i = 0 to 3 **do**
    for j = 0 to 3 **do**
        sum ← sum + $comparisonmatrix_{ij}$
    W_i = sum / 3

B. Scoring the Students

From the above step a relative weight for all the three factors for a specific supervisor is found. Now in the next step we are going to use these weights to find a score for each student. The higher the score is the more desirable a match that student is for the supervisor.

The following steps are followed to score a student.

i. if a student has at-least one common area of interest with the supervisor, he/she gets the total weight added to his/her score. For others no point is added.

ii. The student's number of publication(s) is multiplied with the weight of publication and added to the score.

iii. The student's CGPA is multiplied with the weight of CGPA and added to the score.

iv. The final score is that student's score from that particular supervisor.

Algorithm 2 shows the pseudocode for finding the score of a student by an individual supervisor.

Algorithm 2 Finding Score of a student by a Supervisor

Input: Weight Matrix W =W_1, W_2, W_3

Output: $scoreMatrixStudent_{ij}$ = scrore given to student i by supervisor j

Set sum ← 0, i, j

if student i has one matching research area with supervisor j **then**

sum ← sum+W_1

sum ← sum + Students CGPA * W_2

sum ← sum + Students No. of Publications * W_3

$scoreMatrixStudent_{ij}$ = sum

C. Creating a Score Matrix for Students

Steps A and B are repeated for each and every supervisor to find weight of the factors and the individual scores of the students according to the supervisors. The Scores are stored in a $m \times n$ matrix where m is the number of students and n is the number of supervisors such that.

$$scoreMatrixStudent_{ij} = score\ of\ student\ i\ according\ to\ supervisor\ j$$

D. Normalizing the Score Matrix

In this final step of ranking the students the values of the score matrix are normalized to bring them in the range between 0 and 1 using the following steps -

i. First the highest value in the score matrix is determined.

ii. Next each value of the score matrix is divided by the highest value to find a normalized score matrix.

Algorithm 3 shows the pseudocode for finding the normalized score matrix.

Algorithm 3 Finding Normalized Score Matrix of Students by Supervisors

---

Input: scoreMatrixStudent = set of scores given to all the students by each supervisor

Output: Normalized scoreMatrixStudent

```
Set max ← highest value of scoreMatrixStudent
Set m ← number of students
Set n ← number of supervisors

for i = 0 to m do
        for j = 0 to n do
                scoreMatrixStudent_ij ← scoreMatrixStudent_ij / max
```

Ranking the Supervisors

Similarly while ranking the supervisors the steps below are followed:

A. Finding Weight of the Factors for Students

As it was during the ranking of students, the process of finding the weight of the four factors - area of interest, country, years of experience and number of publications - according to each individual student is almost the same. As the weight of each factor depends on the personal choice of an individual student so the weights are calculated for each of them separately.

The steps followed to find personalized approximate weight of the factors for each student:

i. Each student is asked to put a point (between 1 and 9) for each of the four factors/criteria. Higher the point is the criterion has more priority.

ii. A 4×4 matrix M is created using these values such as

$$M(i, j) = \frac{point \quad given \quad to \quad factor \quad i}{point \quad given \quad to \quad factor \quad j} \tag{3.2}$$

For example, if for a student, supervisors' numbers of publication's point is 5 and area of interest is 9 then the matrix will look like

$$\begin{array}{ccccc}
factors & area \quad of \quad interest & publication & country & years \quad of \quad experience \\
area \quad of \quad interest & 1 & \frac{9}{5} & ... & ... \\
publication & \frac{5}{9} & 1 & ... & ... \\
country & ... & ... & 1 & ... \\
years \quad of \quad experience & ... & ... & ... & 1
\end{array}$$

iii. AHP method is applied on this using the steps described in section 3.3.6.

iv. The final matrix is a 4×1 matrix where each value represents the relative weight between 0 and 1 of the respective factor.

Algorihtm 4 shows the code for finding the weight matrix for an individual student.

Algorithm 4 Finding Weight Matrix for a Student

Input: $FS = FS_1, FS_2, FS_3, FS_4$ = A set of 4 points given to 4 factors by a student

Output: Weight Matrix $W = W_1, W_2, W_3, W_4$

Set $i \leftarrow 0; j \leftarrow 0;$

for i = 0 to 4 **do**
    for j = 0 to 4 **do**
        $comparisonmatrix_{ij} = FS_i / FS_j$

Set T1 $\leftarrow$ $comparisonmatrix_{00} + comparisonmatrix_{10} + comparisonmatrix_{20} + comparisonmatrix_{30}$
Set T2 $\leftarrow$ $comparisonmatrix_{01} + comparisonmatrix_{11} + comparisonmatrix_{21} + comparisonmatrix_{31}$
Set T3 $\leftarrow$ $comparisonmatrix_{02} + comparisonmatrix_{12} + comparisonmatrix_{22} + comparisonmatrix_{32}$
Set T4 $\leftarrow$ $comparisonmatrix_{03} + comparisonmatrix_{13} + comparisonmatrix_{23} + comparisonmatrix_{33}$

for i = 0 to 4 **do**
    for j = 0 to 4 **do**
        if j = 0 **then**
            $comparisonmatrix_{ij} = comparisonmatrix_{ij} / T1$
        else if j = 1 **then**
            $comparisonmatrix_{ij} = comparisonmatrix_{ij} / T2$
        else if j = 2 **then**
            $comparisonmatrix_{ij} = comparisonmatrix_{ij} / T3$
        else
            $comparisonmatrix_{ij} = comparisonmatrix_{ij} / T4$
sum $\leftarrow$ 0;
for i = 0 to 4 **do**
    for j = 0 to 4 **do**
        sum $\leftarrow$ sum + $comparisonmatrix_{ij}$
    W_i = sum / 4

B. Scoring the Supervisors

From the above step a relative weight for all the four factors for a specific student is found. Now in the next step we are going to use these weights to find a score for each supervisor. The higher the score is the more desirable a match that supervisor is for that student.

The following steps are followed to score a supervisor.

i. if a supervisor has at-least one common area of interest with the student, he/she gets the total weight added to his/her score. For others no point is added.

ii. The supervisor's number of publication(s) is multiplied with the weight of publication and added to the score.

iii. The supervisor's years of experience is multiplied with the weight of years of experience and added to the score.

iv. If the supervisor's country matches the desired country of the student then weight of country factor is added to the score too.

v. The final score is that supervisors score from the particular student.

Algorithm 5 shows the pseudocode for finding the score of a supervisor by an individual student.

Algorithm 5 Finding Score of a Supervisor by a Student

Input: Weight Matrix W =W_1, W_2, W_3, W_4

Output: scoreMatrixSupervisor$_{ij}$ = scrore given to supervisor I by student j

Set sum ← 0, i, j

if supervisor i has one matching research area with student j **then**
    sum ← sum+W_1

sum ← sum + Supervisors No. of Publications * W_2
sum ← sum + Supervisors Experience * W_3

if student j's desired country matches with supervisor i **then**
    sum←sum+W_4

scoreMatrixSupervisor$_{ij}$ = sum

C. Creating a Score Matrix for Supervisors

Steps A and B are repeated for each and every student to find weight of the factors and the individual scores of all the supervisors by them. The Scores are stored in a $m \times n$ matrix where m is the number of students and n is the number of supervisors such that.

$$scoreMatrixSupervisor_{ij} = score\ of\ supervisor\ i\ according\ to\ student\ j$$

D. Normalizing the Score Matrix

In this final step of ranking the supervisors the values of the score matrix are normalized to bring them in the range between 0 and 1 using the following steps -

i. First the highest value in the score matrix is determined.

ii. Next each value of the score matrix is divided by the highest value to find a normalized score matrix.

Algorithm 6 shows the pseudocode for finding the normalized score matrix.

Algorithm 6 Finding Normalized Score Matrix of Supervisors by Student

---

Input: scoreMatrixStudent = set of scores given to all the supervisors by each student

Output: Normalized scoreMatrixSupervisor

```
Set max ← highest value of scoreMatrixSupervisor
Set m ← number of students
Set n ← number of supervisors

for i = 0 to m do
        for j = 0 to n do
                scoreMatrixSupervisor_ij ← scoreMatrixSupervisor_ij / max
```

Ranking from both sides

After determining the score matrix for both students and supervisors the two matrixes are added. And to avoid the cases where one party ranks the other too high but himself/herself is too low in that parties rank, the absolute value of the difference between these two values are subtracted from the summation to find the final score matrix *totalScoreMatrix*. $totalScoreMatrix_{ij}$ = $scoreMatrixSupervisor_{ij}$ + $coreMatrixStudent_{ij}$–$absolute(scoreMatrixSupervisor_{ij}$ + $coreMatrixStudent_{ij})$

Algorithm 7 shows the pseudocode for finding the total score matrix.

Algorithm 7 Finding Total Score Matrix

Input: scoreMatrixSupervisor = set of scores given to all the supervisors by each students,
scoreMatrixStudent = set of scores given to all the students by each supervisors

Output: totalScoreMatrix

Set m ← number of students
Set n ← number of supervisors

for i = 0 to m **do**
  for j = 0 to n **do**
    $totalScoreMatrix_{ij} \leftarrow scoreMatrixStudent_{ij} + scoreMatrixSupervisor_{ij} - |scoreMatrixStudent_{ij} - scoreMatrixSupervisor_{ij}|$

Calculating Strength of the Match using TOPSIS Method

The combined matrix found in the previous section shows a two way ranking between the students and the supervisors in which every value represents one individual student's and supervisor's pair strength. As one student's score for a supervisor lies between 0 and 1 and vice versa, so the difference between these two values will also be between 0 and 1, so the values of the total score matrix can be 0 at minimum and 2 at maximum.

According to TOPSIS method described in section 3.3.7, the positive ideal solution or most satisfactory match between a student and a supervisor will be when *totalScoreMatrix* = 2. In this case, both the student and the supervisor have given a score of 1 to each other which is the highest score possible. And because they gave the same score to others the difference between the two scores are 0 making the *totalScoreMatrix* = 2. On the other hand the negative ideal solution or least satisfactory match between a student and a supervisor will be when *totalScoreMatrix* = 0. This situation can occur when both sides gave the other a score of 0 or one of them gave the other a score of 1 but got a 0 from them. The prior indicates from both sides the match is totally undesirable but the latter is also an undesirable match in our system because it only gives full satisfaction to one party but zero satisfaction to the other.

Matching Students and Supervisors using Kuhn-Munkres Algorithm

In this final step Kuhn-Munkres Algorithm is applied to find the stable matching between students and supervisors using the *totalScoreMatrix* using the steps described in 3.3.8. In this case, the highest cost function is used as according to out satisfaction definition the more the *totalScoreMatrix* is the more satisfactory the match will be.

Steps of AHP Method

According to [19], to make a decision in an organised way to generate priorities we need to decompose the decision into the following steps.

i. Define the problem and determine the kind of knowledge sought.

ii. Structure the decision hierarchy from the top with the goal of the decision, then the objectives from a broad perspective, through the intermediate levels (criteria on which subsequent elements depend) to the lowest level (which usually is a set of the alternatives).

iii. Determine the value of each alternative $a_1, a_2, a_3, ... a_n$.

iv. The values are inserted into the comparison matrix D, such that $a_{ij} = \frac{a_i}{a_j}$.

$$D = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}$$

v. Determine D^* such that $a^*_{ij} = \frac{a_{ij}}{\sum_{i=1}^{i=n} a_{ij}}$

$$D^* = \begin{bmatrix} a^*_{11} & a^*_{12} & \dots & a^*_{1n} \\ a^*_{21} & a^*_{22} & \dots & a^*_{2n} \\ \dots & \dots & \dots & \dots \\ a^*_{n1} & a^*_{n2} & \dots & a^*_{nn} \end{bmatrix}$$

vi. The normalized principal Eigen vector can be obtained by averaging across the rows.

$$W = 1/n \begin{bmatrix} p_1 \\ p_2 \\ \cdots \\ p_n \end{bmatrix} \quad \text{Where } p_i = \sum_{j=1}^{j=n} a^*_{ij}$$

The normalized principal Eigen vector is also called priority vector. Since it is normalized, the sum of all elements in priority vector is 1. The priority vector shows relative weights among the things that we compare.

Steps of TOPSIS Method

The basic idea of TOPSIS is quite straightforward. It originates from the concept of a displaced ideal point from which the compromise solution has the shortest distance [110, 111]. Hwang and Yoon [21] further propose that the ranking of alternatives will be based on the shortest distance from the positive ideal solution (PIS) and the farthest from the negative ideal solution (NIS). TOPSIS simultaneously considers the distances to both PIS and NIS, and a preference order is ranked according to their relative closeness, and a combination of these two distance measures.

i. The first step is to construct the decision matrix D, which consists of two sets alternatives, described by:

$$\mathbf{D} = \begin{array}{c} \\ A_1 \\ A_2 \\ \ldots \\ A_m \end{array} \begin{array}{c} \begin{array}{cccc} B_1 & B_2 & \ldots & B_n \end{array} \\ \begin{bmatrix} x_{11} & x_{12} & \ldots & x_{1n} \\ x_{21} & x_{22} & \ldots & x_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ x_{m1} & x_{m2} & \ldots & x_{mn} \end{bmatrix} \end{array}$$

where $A_1, A_2, ..., A_m$ are one set of viable alternatives, and $B_1, B_2, ..., B_m$ are second set of viable alternatives, x_{ij} indicates the rating of the alternative A_i according to alternative B_j and vise versa.

ii. Next the normalized decision matrix $R = [r_{ij}]$ is calculated as -

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{m} x_{ij}^2}} \quad Or$$

$$r_{ij} = \frac{x_{ij}}{x_{ijmax}}$$

The normalized decision matrix R represents the relative rating of the alternatives

iii. Then the positive ideal solutions A^+ (strongest match) and negative ideal solutions A^- (weakest match) are identified.

iv. After that the Euclidean distances from the positive ideal solution A^+ and the negative ideal solution A^- for each alternative A_i and B_j are calculated, as -

$$d_i^+ = \sqrt{\sum_{j=1}^{j=n} d_{ij}^{+2}} \quad and$$

$$d_i^- = \sqrt{\sum_{j=1}^{j=n} d_{ij}^{-2}}$$

v. The relative closeness C_i for each alternative A_i and B_i with respect to positive ideal solution are calculated.

$$C_i^+ = \frac{d_i^-}{d_i^+ + d_i^-}$$

vi. Finally the alternatives are ranked according to the relative closeness. The best alternatives are those that have higher value C_i and therefore should be chosen because they are closer to the positive ideal solution

Steps of Kuhn-Munkres Algorithm

A compact description of the steps of this algorithm, adapted from [2], is given below, and an example is shown in Figure 3.3.

$$\begin{matrix} 10 & 19 & 8 & 15 \\ 10 & 18 & 7 & 17 \\ 13 & 16 & 9 & 14 \\ 12 & 19 & 8 & 18 \\ 14 & 17 & 10 & 19 \end{matrix}$$

Step 1

$$\begin{matrix} 10 & 19 & 8 & 15 & 19 \\ 10 & 18 & 7 & 17 & 19 \\ 13 & 16 & 9 & 14 & 19 \\ 12 & 19 & 8 & 18 & 19 \\ 14 & 17 & 10 & 19 & 19 \end{matrix}$$

Step 2

$$\begin{matrix} 2 & 11 & 0 & 7 & 11 \\ 3 & 1 & 0 & 10 & 12 \\ 4 & 7 & 0 & 5 & 10 \\ 4 & 11 & 0 & 10 & 11 \\ 4 & 7 & 0 & 9 & 9 \end{matrix}$$

Step 3

$$\begin{matrix} 0 & 4 & 0 & 2 & 2 \\ 1 & 4 & 0 & 5 & 3 \\ 2 & 0 & 0 & 0 & 1 \\ 2 & 4 & 0 & 5 & 2 \\ 2 & 0 & 0 & 4 & 0 \end{matrix}$$

Step 4

$$\begin{matrix} 0 & 4 & 0 & 2 & 2 \\ 1 & 4 & 0 & 5 & 3 \\ 2 & 0 & 0 & 0 & 1 \\ 2 & 4 & 0 & 5 & 2 \\ 2 & 0 & 0 & 4 & 0 \end{matrix}$$

Step 5

$$\begin{matrix} 1 & 5 & 2 & 3 & 3 \\ 1 & 4 & 1 & 5 & 3 \\ 3 & 1 & 2 & 1 & 2 \\ 2 & 4 & 1 & 5 & 2 \\ 3 & 1 & 2 & 5 & 1 \end{matrix}$$

Step 6

$$\begin{matrix} 0 & 4 & 1 & 2 & 2 \\ 0 & 3 & 0 & 4 & 2 \\ 2 & 0 & 1 & 0 & 1 \\ 1 & 3 & 0 & 4 & 1 \\ 2 & 0 & 1 & 4 & 0 \end{matrix}$$

Step 7

$$\begin{matrix} \boxed{0} & 3 & 1 & 1 & 1 \\ 0 & 2 & \boxed{0} & 3 & 1 \\ 3 & 0 & 1 & \boxed{0} & 1 \\ 1 & 2 & 0 & 3 & \boxed{0} \\ 3 & \boxed{0} & 1 & 4 & 0 \end{matrix}$$

Step 9

Figure 3.3: Steps of Kuhn-Munkres Algorithm

1. Step 1: Arrange the information in a two dimensional matrix.
2. Step 2: Ensure that the matrix is square by the addition of dummy rows/columns if necessary. Conventionally, each element in the dummy row/column is the same as the largest number in the matrix.
3. Step 3: For each row of the matrix, find the smallest element and subtract it from each element in its row.
4. Step 4: If there are columns without a zero, reduce the columns by subtracting the minimum value of each column from that column.
5. Step 5: Cover the zero elements with the minimum number of lines it is possible to cover them with.
6. Step 6: Add the minimum uncovered element to every covered element.
7. Step 7: Subtract the minimum element from every element in the matrix.

8. Step 8: If the number of lines covering the zero elements is not equal to the number of rows, return to step 6.

9. Step 9: Select a matching by choosing a set of zeros so that each row or column has only one selected.

Apply the matching to the original matrix, disregarding dummy rows.

Correctness of the Solution

To find that our solution works better than others lets first assume that the best system for the first step, weight measurement would be to use the pure AHP method which is to pairwise compare between different alternatives or choices. So if we asked the students or supervisors how they compare each criterion with comparison to every other criteria the following problems would occur

A. The number of calculation to calculate the weight matrix for a single student or supervisor will be very high. As there will be 10 comparison for four criteria and six when there are three criteria. So the total number of calculation to find all the weights will be huge.

B. Secondly, as the students or supervisors will be asked to compare two criteria, say, area of interest and number of publication - how one is important compared to the other - in the scale of ten stretched both side of zero, it will be harder for them to provide a comparison this way because it is complicated to give a relative comparison between a pair of thing.

Because of these drawbacks, it is a better idea to ask them to put a numeric value between 1 and 9 for each criterion. For it is easier for a person to put a relative value on a scale for a single object. So, we believe, it is more effective to measure the weight of the criteria using point scale rather than pairwise comparison.

In the next step, when we are calculating the score for each supervisor and student according to each student and supervisor respectively for some values we are grading the attribute using a predefined value to scale the values in a range. For example, a supervisor has 100 publications where another supervisor has 10. If we take these exact values and multiply them with the weight each student assigned to the 'no. of publication' criteria the first supervisor will have a much higher score than the second one. In fact, doing this will put the supervisors who has a very high number of publications in the top score range even for the students who puts the lowest weight on the 'no. of publication' criteria. It'll ultimately result in faulty calculation as these supervisors will be every students' top choice and those who're not paired with any of them will be shown as the unsatisfied pairs. The same is true for 'years of experience' criteria too.

So, in this aspect it is better to grade the properties of each criterion to a specific range to treat them equally during the score calculation. This way the scores will reflect a student or supervisor's true preference perfectly.

In the third step, scaling the score matrix into values between 0 and 2 actually gives a uniform value to all the matrix elements. This way it's ideal to define a satisfactory matrix where 2 defines the highest satisfaction between a pair and 0 defines the lowest satisfaction. Without scaling the values, the matrix will have different values at each run and for different set of students and supervisors. So, we would have to define a different satisfaction measurement for each run. Also the satisfaction measurement decided in this way wouldn't have been able to give a uniform and robust pair-strength calculation strategy in this way.

From, above discussion, it can be said that, for a sound system with less time and calculation complexity the modification we used in our method works best.

Analysis of the Proposed Method

Consider a situation, where a student enters the system and enters his/her criteria for finding a suitable match. He/she puts a weight on each of the criterion so that the top most important factor gets the highest point and so on. These weights are then used to find relative weights of each of the criteria. After that each supervisor is weighted using these weights so that everyone of them is assigned a score by that student. The supervisor who has the highest score will be the most satisfactory match for the student.

After that the student was then scored by every supervisor according to their preference and weights they put on the criteria for choosing a student to supervise and using the information provided by the student about his/her research interest, CGPA etc. Using these both way scores are then fed to find the matching pairs by feeding it to Kuhn-Munkres Algorithm.

This procedure is followed whenever the set of students and supervisors are changed i. e.

a new student or supervisors has entered into the system or has been removed from it. So in this way each time the set of pairs selected in such a way minimizes the dissatisfaction of those matched pairs from that very given set of students and supervisors.

Conclusion

In this chapter, the algorithms and steps of finding the weight matrix and score matrix for each student and supervisor are described. These score matrix are then used to create a combined score matrix. From this combined matrix a strength function is defined to determine the amount of satisfaction between each pair of student-supervisor. Finally the assigned pairs are selected from the matrix in such a way that minimizes the total dissatisfaction of all the matched pairs.

Chapter 4

Experimental Result

In this chapter we are going to discuss about our experimental results in a simulated environment. At First, we discuss the experiment setup and then compare the performance of the solutions with other matching schemes.

Machine Configuration

The simulation environment is designed using the programming language JAVA. JAVA is a platform independent language which can be run in a cross-platform computing environment. The machine needs to have the java runtime environment (jre) installed on it. The minimum machine configurations required for running our simulation is the following -

i. Platform(s): Windows 7/macOS

ii. Java version: 7.0

iii. RAM: at-least 2 GB

iv. IDE: Netbeans 7.4

Data Collection

To test out our proposed system, we needed specific data from both the students and the supervisors. This data basically concerns the choices of the students and supervisors about their personal preferences while looking for a supervisor or selecting to mentor a student respectively. We tried to find out that even if the students do not know the supervisors personally, what will be their ideal thesis supervisor's qualities. On the other hand to the supervisors we asked the question about what kind of students they would like to mentor. For our thesis, we used three main sources of data. Firstly, we conducted a survey among the fourth year students and faculty members of CSE department of a private university in Dhaka, Bangladesh. Secondly, we used the data collected by Humayara Binte Rashid and Annita Tahsin Priyoti on the thesis students of CSE department of BUET. Thirdly, we used an online data source [112] which contains data where students listed their preferred subject areas in order of preference. This dataset also contains subject areas with related supervisors who supervise in that area. Lastly, we used some manipulated data to find out results in different situations. This sample set contains the students in their final year who has the necessary knowledge about their subject areas and also has a has the idea on what outcomes they want from their research. For the supervisor, all of them has conducted under-grad thesis with their students in the recent years. There were 120 students and 16 supervisors that we used for our data set. And they represent an average set of students and supervisors where the student data set contain student with varying CGPA's and other qualities. And the supervisors range from Professors to lecturers with different research preferences and ideas.

Creating A Scenario

During the data collection we created two separate questionnaires for the students and the supervisors. From the students the information collected is how they score each category/factors while looking for a suitable supervisor. At the same time, the students are asked to provide some of their academic details as we need this information to rank them according to the supervisors. Similarly the questionnaire for the supervisors gathers data about their academic qualities to be used in their ranking by students besides the data about their preference of factors when deciding on a student.

Figure 4.1 shows a sample questionnaire for students and Figure 4.2 shows a sample questionnaire for supervisors.

Name of the student: ....

Institution: ....

Current CGPA: ....

Research Interests (topics/fields): ....
....
....
....

Number of Publications (if any): ....

Approximate weight of the following factors when looking for a supervisor

Instructions: Please assign a score between 1 to 9 for each of the factors below. How much valuable you find the criteria while looking up a supervisors profile should reflect in the score (9 being the highest importance and 1 being lowest). The criteria are shown in no particular order.

1. Research Interest of the Supervisor
2. Years of Experience of the Supervisor
3. Number of Publications of the Supervisor
4. Country/Institute of the Supervisor

Figure 4.1: A Sample Questionnaire for Students

Name of the supervisor: ....

Institution: ....

Experience (Years): ....

No. of Publication (if any): ....

Research Interests (topics/fields): ....
....
....
....

Approximate weight of the following factors when accepting for a student

<u>Instructions</u>: Please assign a score between 1 to 9 for each of the factors below. How much valuable you find the criteria while reviewing a students' application should reflect in the score (9 being the highest importance and 1 being lowest). The criteria are shown in no particular order.

1. Research Interest of the Student ☐
2. CGPA of the Student ☐
3. Number of Publications of the Student ☐

Figure 4.2: A Sample Questionnaire For Supervisors

After collecting the data, they are fed into the system. As the system works on a fixed number of students and supervisors, the output of the system is a set of matched student-supervisor pair, is used to calculate the satisfaction of the students and supervisors. A student matched with a supervisor who matches most of his criteria will be a satisfactory match for him. On the other hand, a student matched with a supervisor who matches least to none of his/her criteria will be disappointed by the match. The same is true for a supervisor too. From looking at the factors it can be said that a student with high CGPA or high number of published work has more chance in getting a satisfied match than other students. For the supervisor the distinction is not so profound. As where a student may be interested in an experienced supervisor, there another student can just ignore this factor and focus more on a matched research area. On the other hand, to find a suitable match that satisfies both parties is more of a challenge. Where one side may rank the other side high, for the second side the match may not be so desirable. So the biggest challenge is to find a match that satisfies both as closely as possible. The more matches that satisfy both sides the better the ranking system will be.

Performance Measurement Matrices

In our experiment, the higher the ranking of a student or supervisor that one is matched with, the more is the satisfaction. There are three angles from where the matches' satisfaction can be measured - a) Satisfaction of the Student, b) Satisfaction of the Supervisor and c) satisfaction of the matching. These matrices are defined below.

Satisfaction of the Student

When a student gets assigned with the supervisor who is ranked high on his/her list then it'll be a satisfactory match for him/her. How many students are getting assigned to the supervisor on their priority list is a way to measure how the system performs.

Satisfaction of the Supervisor

In the same way when the supervisor gets the student who fulfils his/her criteria most will be a satisfactory assignment for him/her. The percentage of supervisors who gets assigned to their upper choices is a measurement of the performance too.

Satisfaction of the Matching

In a matched pair, how much both side is satisfied is also a measure to calculate the performance. For, if a student ranks a supervisor very high, but the supervisor ranks the same student low on their list then the assignment would be a high satisfaction match for the student and low satisfaction match for the supervisor. Such a match is not desirable as it is not a fully satisfactory match for both parties. So, the third measurement of performance is how much pair-wise satisfactory match does the system generates.

Experimental Results with Analysis

In this section, first we are going to look at the experiments done by other researchers in relevant situations and then we are going to show the experiment results with comparison of the performance graphs of each experiment.

Some Relevant Experiments

A large number of problems that arise in academic and industrial areas have several conflicting objectives that need to be optimized simultaneously [62]; they are called multi-objective optimization problems (MOPs). The most commonly adopted notion of optimum in multi-objective optimization is Pareto optimality, which refers to finding the best possible trade-offs among the objectives of a multi-objective problem. These trade-off solutions constitute the so-called Pareto optimal set. The image of the Pareto optimal set is called the Pareto front. Among the different techniques available to solve MOPs, multi-objective evolutionary algorithms (MOEAs) have become very popular, mainly because of their flexibility and ease of use. Modern MOEAs normally aim at producing, in a single run, several different solutions, which are as close as possible to the true Pareto front [62]. For several years, MOEAs adopted a selection mechanism based on Pareto optimality. However, in recent years, it was found that Pareto-based MOEAs cannot properly differentiate individuals when dealing with problems having four or more objectives (the so-called many-objective optimization problems [63]). This has motivated the development of alternative selection schemes from which the use of performance indicators has been (until now) the most popular choice [113]. When using indicator-based selection, the idea is to identify the solutions that contribute the most to the improvement of the performance indicator adopted in the selection mechanism.

On the other hand, MOEAs based on decomposition have also become popular in recent years. Perhaps, MOEA/D is the most popular MOEA based on decomposition. This algorithm decomposes the MOP into N scalar optimization sub-problems and it solves these sub-problems simultaneously using an evolutionary algorithm. MOEA/D has shown to be a good alternative to solve MOPs with low or high dimensionality (regarding objective function space). However, MOEA/D has two important disadvantages. The first is that it generates a new solution from a unique neighborhood, i.e., the new solution cannot be generated from individuals of different neighborhoods. And, the second is that a new solution with a high fitness can replace several solutions, and then, the population can lose diversity. Li and Zhang proposed in [114] a variant of MOEA/D and they called it "MOEA/D-DE". This proposal allows that a new individual will be generated from individuals of different neighborhoods. Also, it restricts the number of solutions that can be replaced by the same individual. However, both proposals MOEA/D and MOEA/D-DE generate a new solution, and then, they look in which sub-problem the new solution is better than the current solution but they do not consider the case where the solution which was replaced could improve the solution of another sub-problem, i.e, both algorithms assign the best individual to each sub-problem in an independent way, without considering the best assignment globally.

On the other hand, In [10] Kuhn-Munkres Algorithm is used to solve Sailor Assignment Problem(SAP). But the Kuhn-Munkres Algorithm is defined only for a single objective function, whereas SAP requires the simultaneous consideration of multiple objectives (training score, permanent change of station cost, commander choice, and sailor choice). To resolve this incompatibility, single objective instances of SAP are obtained using weight vectors, with the resulting goal to optimize $w_1 * TS + w_2 * PCS + w_3 * SR + w_4 * PCS$ with $w_i \in [0, 1] and \quad w_1 + w_2 + w_3 + w_4 = 1$. To obtain a diverse set of Pareto optimal solutions, the single objective problem is solved for each one of the weight vectors obtained via recursively subdividing the weight space.

But, in this implementation the weights are fixed and predefined. However, in our system we didn't use any predefined or fixed weight. Rather we calculated the weightage of the factor dynamically from the specific dataset that the system is working on.

Results and Analysis of our Experiments

As the data were collected from three different sources, there is a bit of bias among the results. So the result we now discuss is an average one. Now we will discuss different stages of our experiment and the results –

Measuring Weights of the Criteria

The first step of our system is measuring weights of each of the criterion according to every individual student and supervisor. This step is the most important step of the experiment as calculating the weight is detrimental in ranking the students and supervisors and their subsequent matching. The measurement of satisfaction is defined by the score calculated from these weights so if the students or supervisors are not careful in putting a relative value on each criteria it will cause faulty ranking and eventually unsatisfactory assignments.

Satisfaction of the Students

The satisfaction of the students depends on how they prioritise the criteria important to them.

When a Student puts Same Weight on Each Criterion

If a student gives the same emphasis on all the criteria (area of research, no. of publication, experience, country/institute) then ranking the supervisor is a straight forward experiment. In this case a supervisor who has matching area of interest, high number of publications, longer experience and comes from the students' desired country/region is going to be ranked higher. Next in the rank will appear supervisors who share the same area of interest and come from the desired country/institution but have less number of publications and/or less experience. Those supervisors who either have same area of interest or come from the desired country/region but not both will be next on his/her ranking. And at the bottom of the ranking will be supervisors who neither share the same research area nor come from the desired country/institute as per the student.

When all the Students has Same First Priority

In case of scenarios where all the students put highest emphasis on the same criterion will affect the ranking in different ways depending on the criterion.

Highest Emphasis on Area of Interest Different students have different areas of interests, and so are the supervisors. Hence when all the students place their highest points in area of research interest category the chances of the students ending up with their top choice supervisors are very high. In our experience, about 80% of the students ends up with their most favourable supervisor when all of their first priorities are matching area of interest with the supervisor.

Figure 4.3 shows the satisfaction in matching if the students put Area of Interest as their first priority.

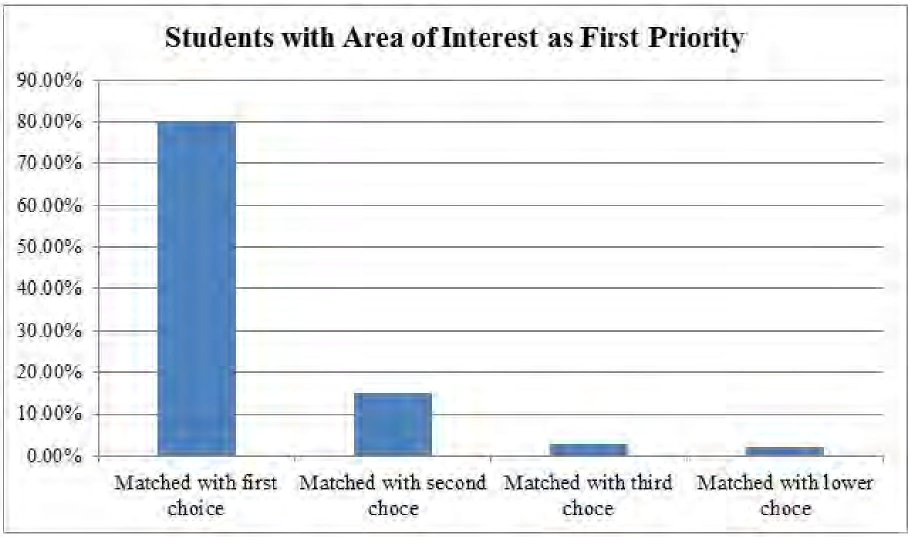

Figure 4.3: Students with Area of Interest as First Priority

Highest Emphasis on Supervisor's Number of Publications If all of the students are most interested in how much publication the supervisor has then supervisors with a large number of publications become the first choice for most of the students. In this case the number of students ending up with their number one choice comes down to around 60%. Those students with a better CGPA, published works and who matches the area of interest with the desired supervisors has the highest chance of getting matched with their first or second preference in this case.

Figure 4.4 shows the satisfaction in matching if the students put Supervisor's Number of

Publications as their first priority.

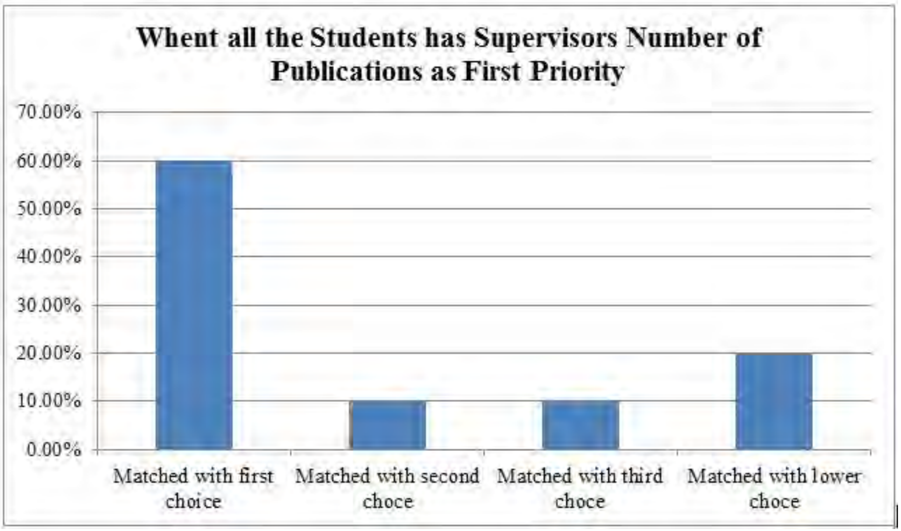

Figure 4.4: Students with Supervisor's Number of Publications as First Priority

Highest Emphasis on Supervisor's Experience As with the previous scenario, if most students prefer supervisors with more experience under their belts, it will yield almost the same result. In this case too around 60% students get matched with their first or second choice.

Figure 4.5 shows the satisfaction in matching if the students put Supervisor's Experience as their first priority.

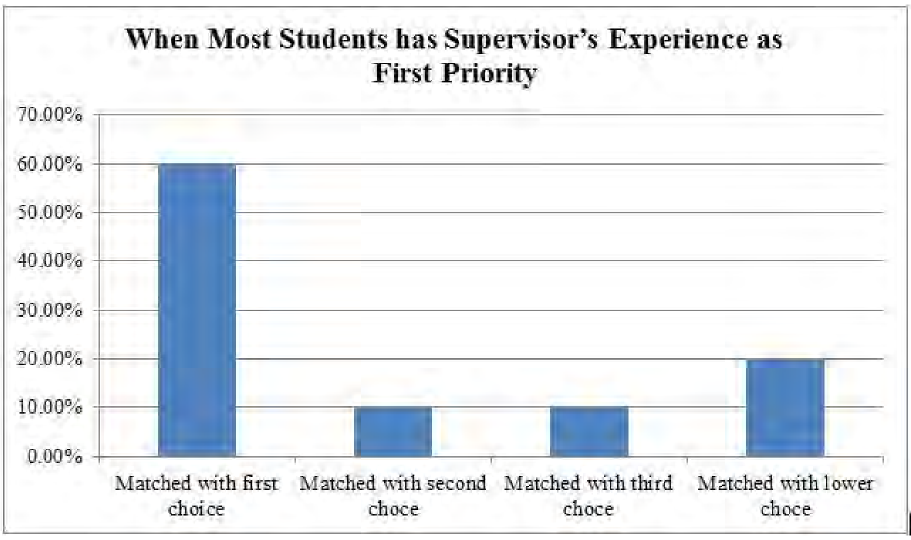

Figure 4.5: Students with Supervisor's Experience as First Priority

When Different Students has Different First Priorities

This is basically a real world scenario where different students have different first priorities. This situation also produces the best result when matching the students with supervisors. In this case more than 90% of the students will be chosen to match with their first or second most favourable supervisor. With a large number of students and supervisors, no student, in this case will end up with supervisor who is not in his/her top 5 ranks.

Figure 4.6 shows the satisfaction statistics in matching if the students have different first priorities.

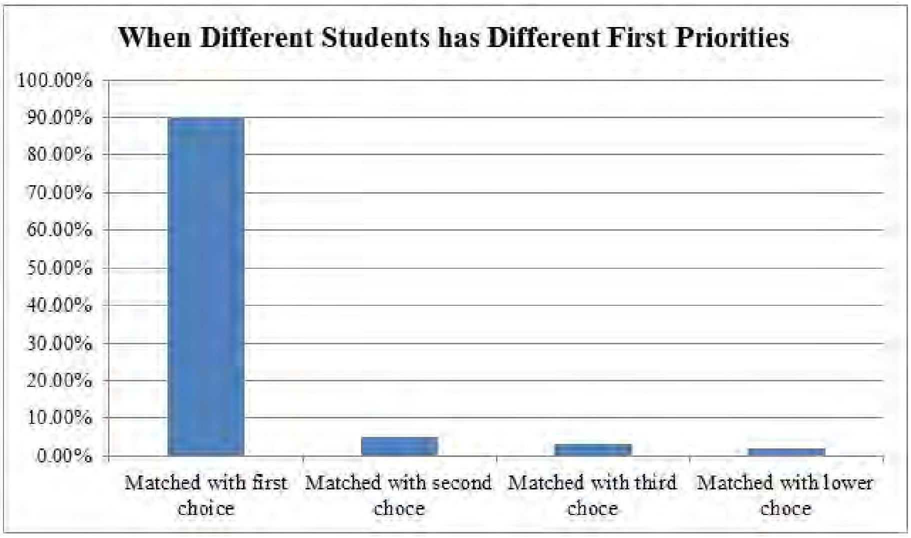

Figure 4.6: Different First Priority for Different Student

Satisfaction of the Supervisor

The satisfaction of the supervisors depends on how they prioritise the criteria important to them.

When a Supervisor puts Same Weight on Each Criterion

If a supervisor finds all the three criteria (CGPA of the student, number of publications, area of interest) equally important then in his/her ranking a student with high CGPA, same area of interest and previous published work will be placed on the top. Next a student with same area of interest who already has some published work will come. A small discrepancy in CGPA on the lower side will not affect the ranking much if the student already has publications.

However, a student with low CGPA but one or more published work will be ranked higher than a student who has high CGPA but no publication in this case. And finally lowest in the raking here will be students with low CGPA, no publications and no shared research area with the supervisor.

When Every Supervisor has Same First Priority

In case of scenarios where all the supervisors put highest emphasis on the same criterion the effect on the ranking will be different depending on the criterion.

Highest Emphasis on Area of Interest Each supervisor has different research areas that they usually work and students would love to work on. So when the supervisors decide to accept students mainly on the basis of a shared area of interest, it is easy to match them with their desired students. As any student and supervisor who has same subject interest will be perfect matches for each other. In this situation the chances of the supervisors getting students from the top of their list is more than 80%.

Figure 4.7 shows the satisfaction in matching if the supervisors put Area of Interest as their first priority.

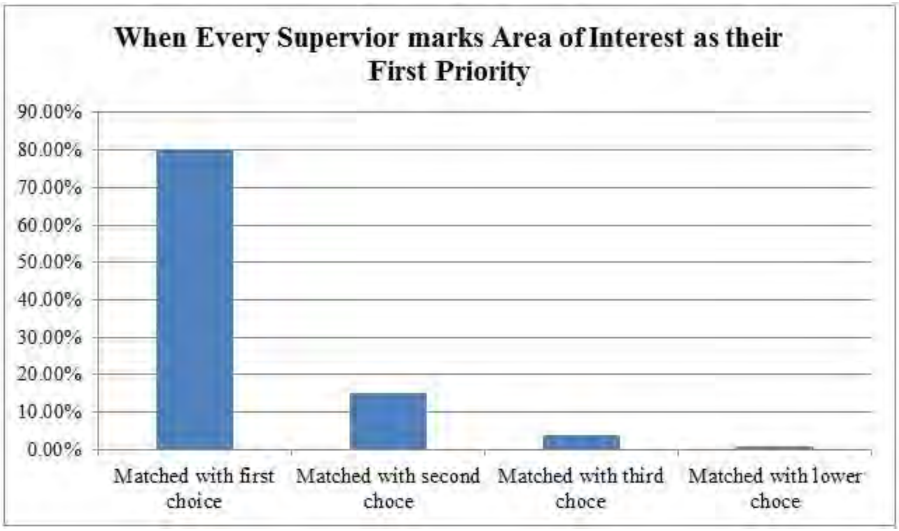

Figure 4.7: Supervisors with Area of Interest as First Priority

Highest Emphasis on Student's CGPA When the supervisors are only keen on the students' CGPA, generally the student with high CGPAs fills their top-list. In this situation most supervisors will share the same first or second priority students. Supervisors with matched area of interest or high number of publications are more likely to be matched with these desired students. The chances of a supervisor getting his/her first or second choice student is around 60% in this situation

Figure 4.8 shows the satisfaction in matching if the supervisors put students CGPA as their first priority.

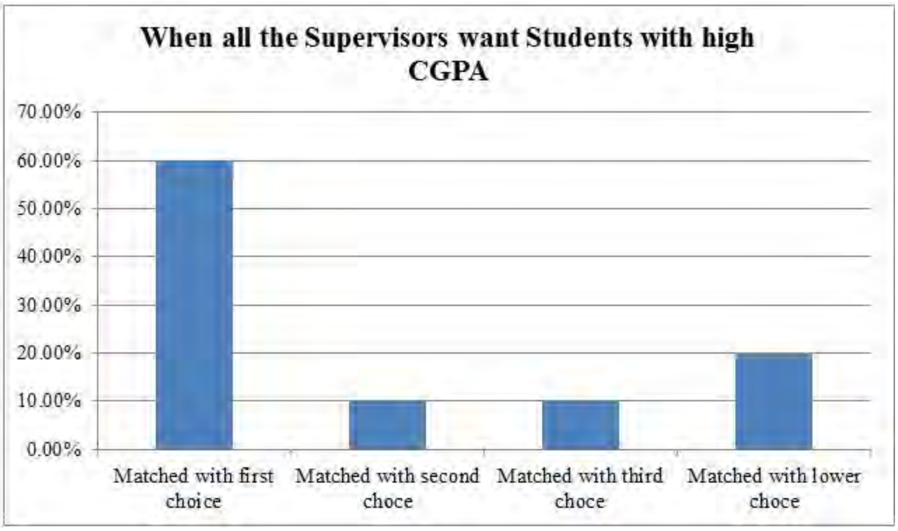

Figure 4.8: Supervisor with Students' CGPA as First Priority

Highest Emphasis on Student's Number of Publications Students generally have few to no publication during their undergrad years. So if supervisors place highest points on this criterion then the number of students on all of their top-list in even fewer and the chances of them getting their first or second priority student is less than 50% in this scenario.

Figure 4.9 shows the satisfaction in matching if the supervisors put student's no. of publications as their first priority.

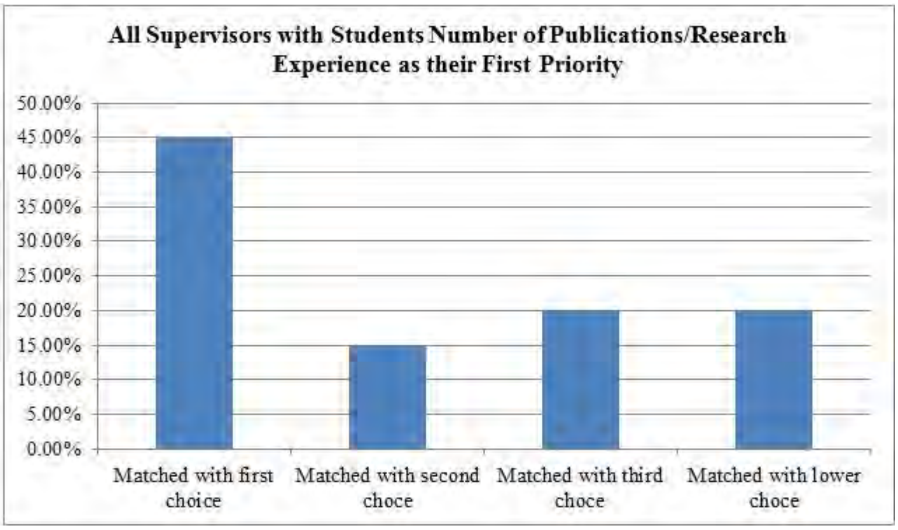

Figure 4.9: Supervisor with Students' no. of publications as First Priority

When Different Supervisors has Different First Priorities

As it is with the students, for too the supervisors this is an ideal scenario which works the best way when matching them with students. In this case more than 90% supervisor will be chosen to match with their first or second priority student. With large number of students and supervisors, no supervisors, in this case will end up with students who are not in his/her top 5 ranks.

Figure 4.10 shows the satisfaction in matching when the supervisors have different first priorities.

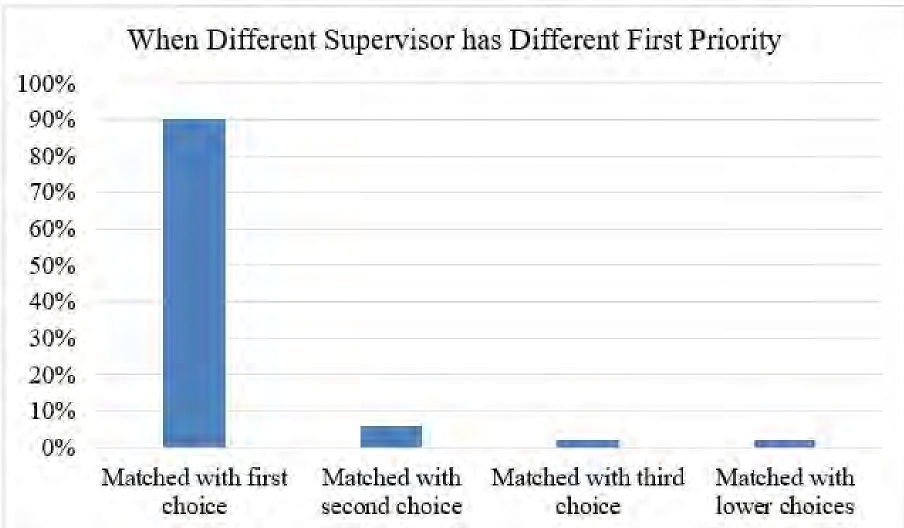

Figure 4.10: Supervisor with Different First Priorities

Satisfaction of the Match

In an ideal situation a student and supervisor paired up with each other when both are each other's first priority would be the perfect satisfaction for both of them. But in our experiment, we have seen, sometimes it is a better match when in a student-supervisor pair the student is the first priority of the supervisor but the supervisor is not the first choice of the student or vice versa. So in the final matching a matched pair where both hold a close to top position in the others list will be a desirable match. Generally, using this ranking, around 90% of the pairs is such where each side has a counterpart from their top three choices.

Comparison with Other Algorithms

Our solution uses a variation of the AHP method to find out the weights of different criteria for both students and supervisors. To prioritize a criterion among various criteria this is the most widely used method. On the other hand TOPSIS is recognized for its ability to find the solution closest or farthest from the ideal solution. Our use of TOPSIS for finding out the solution closest to the ideal solution is also robust is this sense.

Lastly to find out the desired matches that maximises the satisfaction of students and supervisors were the real challenge. There are several algorithm used over the years to find matching in similar situation. We used the Hungarian Algorithm to achieve the task in our system.

Here we will discuss how Hungarian Algorithm fares better than other algorithms for the current problem-

Stable Marriage Algorithm/Gale-Shapely Algorithm

The most widely used matching algorithm is the Gale-Shapely algorithm. But it has some major shortcomings. Namely, this algorithm is biased to one side of the matching. From which side the matching is being conducted will dominate the final matched pairs where this specific side is more likely to end-up being matched with their desired partner than the other side. This bias continues to the sequence of elements too. In some cases, being in a better position will mean getting a partner from the top of their list.

The graph in Figure 4.13 below shows that the possibility of students ending up with their first choice, when SM Algorithm is used from their perspective is 70-75% but for supervisors it is below 50%. Same is true when the SM is applied from supervisor's perspective; more than 70% supervisors will end up with their first choices whereas the number of students getting their first choice is well below 70%. Both statistics are taken when the students and supervisors have different preferences. But the number will fall below 20% for both cases when the supervisors and students all focus on same type of criteria and rank one kind of profile above all other.

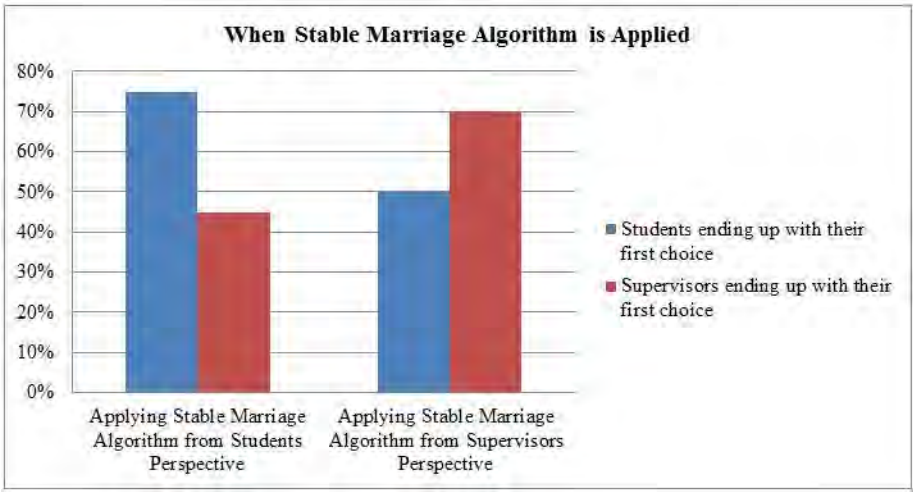

Figure 4.11: When Stable Marriage Algorithm is Applied

Maximum Bipartite Matching Algorithm

In a complete bipartite graph to find the maximum-weight matching this algorithm is used. The Hungarian algorithm is a version of this algorithm. But this algorithm has the drawback that there must be equal number of vertices on both sides. This problem can be overstepped using dummy or fake nodes to match the number of nodes on both sides. This is actually what Hungarian algorithm does. As our problem does not guarantee an equal number of students and supervisors it was better for us to use Hungarian Algorithm than Maximum Bipartite Matching Algorithm.

Student-Project Allocation

This algorithm is closer to our solution in the sense that it does not require equal number of projects and student to match them. A project can have more than one student but not vice versa. But this algorithm follows the same path as the SMP and does not guarantee maximum satisfaction among all the matched pair for a given data-set. Also using SPA algorithm for our case does require some extra modification. If we try SPA algorithm from students' perspective where students with better CGPA gets allocated with their chosen supervisors first then about 60% of the students will end up with their chosen supervisor, but 40% of the supervisor will get the first student they want. On the other hand if the SPA is applied from supervisors perspective where most senior supervisor get their picks first only 55% of the Supervisors will get their first choices, but from students point of view about 30% will get matched with their most desired supervisor. Similarly, if we choose another criteria number of publications of the supervisors and let the supervisors with most number of publications get matched before others then about 70% of the supervisors will get their chosen students but from student's perspective the number will be around 40%. Finally, if we only take the area of interest into consideration, then about 70% students and supervisors will end up with their first choices.

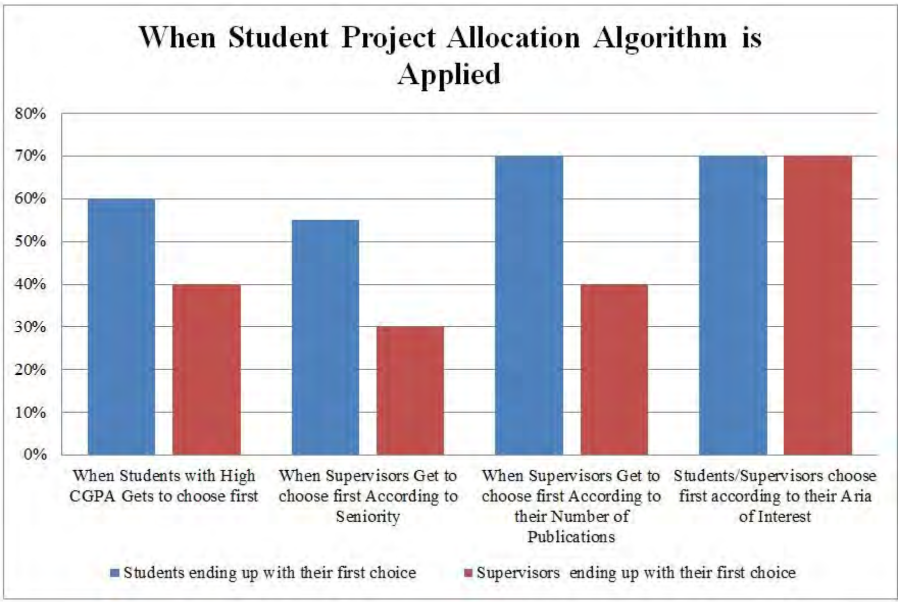

Figure 4.12: When Student Project Allocation Algorithm is Applied

From the above argument, we decided to use Hungarian Algorithm to find out final matches as it not only assure matching of every student and supervisor but also guarantees their most possible satisfaction too.

Comparison with Similar Solutions

There are different criteria about how a system can be measured. For example, most of the previous works focus on run-time and the amount of satisfaction to compare with other solutions. As for our problem, it focuses on the satisfaction of both students and supervisors. So comparing it with other works will be a tough job. As most of the previous works basically focused on students' perspective. We have seen in the previous section that using most of the other matching algorithms is not suitable when we focus on both students and supervisors satisfaction. The solution can be implemented using GA too. Which is how it is done in the paper [15], where they focused on the satisfaction of both students and supervisors by implementing GA while matching the pairs. They try to increase the strength of the matching by swapping students among supervisors in each iteration and if the new generation is stronger than the old one the solution set is taken to be used in the next iteration. When there is no improvement in the solution in 20 consecutive iteration the system decides that this should be the final matching. This system assures 90% satisfaction for both students and supervisors after the final iteration. But though the satisfaction level is pretty high, the run-time is even higher. And as the number of students increase the time complexity also increases. So for a huge data set this system will take a large amount of time to reach the final combination. On the other hand, our system assures 90% satisfaction for both students and supervisors in an ideal situation. And also assures the maximisation of the collective satisfaction of all the pairs finally matched together.

Figure 4.13 shows the relative comparison of the runtime with our solution

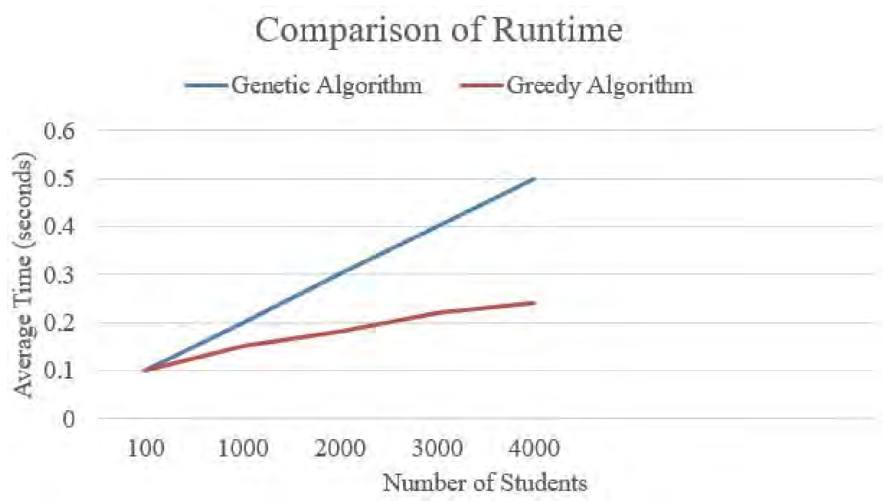

Figure 4.13: Comparison of Runtime

Summary and Conclusion

From the above discussion, it can be said that, using the AHP method for finding out relative importance of the factors that influences both the student and supervisors can lead to an effective method to find a subsequent score for each pair of student-supervisor. After defining the strength of this score to determine the satisfaction of the specific assignment between that student and the supervisor can be calculated using the TOPSIS method. Lastly using these scores the set of assignments are calculated such that the total dissatisfaction is minimized.

Chapter 5

Conclusion and Future Work

Assignment problem has gained a lot of attention in the last few decades. They have different uses in different sectors such as computer science, economics, finance, education, job assignment etc. Various matching algorithms has been developed to handle these scenarios. Especially these algorithms have widely been used in student-supervisor assignment problems, notably, in thesis or project supervision assignment for undergrad student. And even though these solutions has some drawbacks for example, they are either student or supervisor/project optimal, or that they need a clear preference list to work out the matching pairs, they generally works best for single objective functions. But when we consider both the students and projects/supervisor with no clear-cut preference list it becomes difficult to match the pairs. Moreover, when we consider the preference objectives for both parties the problem becomes a multi-objective function. To solve this problem using the matching algorithms we need to convert them into a single objective problem with straight priority list from both sides.

Summary

In this thesis, we have tried to find a ranking system for students and supervisors when they are looking for a matching for post-grad studies. As in this case both the students and supervisors have no personal idea about the other and hence no straight preference list. During these cases the students and supervisors generally depend o their preferred factors to sort out among the supervisors and students respectively. Those who matches their priorities the most are the best matches for them. At first, we tried to find out the factors that influences the students and the supervisors individually. From these factors we found a relative weight for each of the factors for each student and supervisor separately. Using these weights we have created a ranking for the students where every student has a list of all the supervisors catalogued from high to low preferences according to them. Similarly, each supervisor also has a similar ranked list of students using their distinguished weights. Taking these two rank matrices we calculated a third matrix that show the matching strength between every pair of student and supervisor, thus making the problem a linear single function problem. These preference lists are then fed into the Hungarian algorithm method to find the suitable matches. During the simulation, we have fed the system with different student and supervisor preferences to determine the strength of the ranking system.

Future Works

There are some scopes of this system for future work. They are:

i. The system can be similarly extended for students and specific post-grad programs where students preference and program requirements are given.

ii. The system can be used in problems like job searching, marriage match making or similar situations where both party is unknown to the other but has some preferred qualities.

Bibliography

[1] B. Ebert, “Stable matching, as a game.” https://towardsdatascience.com/ stable-matching-as-a-game-a68c279d70b, September 2021.

[2] W. M. Herbert, J. R. Kenneth, and M. Andrew, “Phd students evaluations of research supervision,” *The Journal of Higher Education*, vol. 73, no. 3, pp. 313–348, 2002.

[3] L. McAlpine and J. Norton, “Re-framing out approach to doctoral programs: An integrative framework for action and research,” *Higher Education Research Development*, vol. 25, no. 1, pp. 3–17, 2006.

[4] G. Ives and G. Rowley, “Supervisor selection or allocation and continuity of supervision: Phd students progress and outcomes,” *Studies in Higher Education*, vol. 30, no. 5, pp. 535–555, 2005.

[5] M. Niederle, A. E. Roth, and T. Sonmez, “Matching and market design, in s. derlauf and l. blume (eds.),” *The New Palgrave Dictionary of Economics2nd edn. (Palgrave-Macmillan*, 2008.

[6] A. E. Roth and X. Xing, “Jumping the gun: imperfections and institutions related to the timing of market transactions,” *American Economic Review*, vol. 84, no. 4, pp. 992—-1044, 1994.

[7] A. E. Roth, “The evolution of the labor market for medical interns and residents: a case study in game theory,” *Journal of Political Economy*, vol. 92, no. 6, pp. 991—-1016, 1984.

[8] “The match, national resident matching program.” http://www.nrmp.org, September 2020.

[9] C. Olson, “Is your institution user-friendly? essential elements of successful graduate student recruitment,” *College and University*, vol. 67, no. 3, pp. 203–214, 1992.

[10] D. Dasgupta, G. Hernandez, D. Garrett, P. K. Vejandla, A. Kaushal, and R. Yerneni, “A comparison of multiobjective evolutionary algorithms with informed initialization and kuhn-munkres algorithm for the sailor assignment problem,” *Proceedings of the 10th annual conference companion on Genetic and evolutionary computation*, pp. 2129–2134, 2008.

[11] D. Gale and L. S. Shapley, “College admissions and the stability of marriage,” *American Mathematical Monthly*, vol. 69, no. 1, pp. 9–14, 1962.

[12] D. J. Abraham, R. W. Irving, and D. F. Manlove, “Two algorithms for the student-project allocation problem,” *Discrete Algorithms*, vol. 5, no. 1, pp. 73–90, 2007.

[13] H. O. Salami and E. Y. Mamman, “A genetic algorithm for allocating project supervisors to students,” *International Journal of Intelligent Systems and Applications*, vol. 8, no. 10, p. 51, 2016.

[14] D. H. Ratna, A. Syaiful, and Marjono, “Allocation of thesis supervisor using genetic algorithm,” *Journal EECCIS*, vol. 12, no. 1, pp. 26–32, April 2018.

[15] S. Victor, C. Rithin, A. Reyhan, and J. Vicente, “A near pareto optimal approach to student-supervisor allocation with two sided preferences and workload balance,” *Applied Soft Computing*, vol. 76, no. 1, pp. 1–15, March 2019.

[16] J. D. Garrett, J. Vannucci, R. Silva, D. Dasgupta, and J. Simien, “Genetic algorithms for the sailor assignment problem,” *Proceedings of the 2005 Genetic and Evolutionary Computation Conference (GECCO-05)*, pp. 1921–1928, 2005.

[17] A. Anwar and A. Bahaj, “Student project allocation using integer programming,” *IEEE Transactions on Education*, vol. 46, no. 3, pp. 359–367, 2003.

[18] M. Chiarandini, R. Fagerberg, and S. Gualandi, “Handling preferences in student-project allocation,” *Annals of Operations Research*, vol. 275, no. 1, pp. 39–78, 2019.

[19] T. L. Saaty, “The analytic hierarchy process: Planning, priority setting, resource allocation,” *McGraw-Hill*, 1980.

[20] J. L. Matthew and R. L. Nydick, “Group decision making in higher education using the analytic hierarchy process,” *Research in Higher Education*, vol. 38, no. 5, pp. 593–614, 1997.

[21] C. L. Hwang and K. Yoon, “Multiple attribute decision making: Methods and applications,” *Springer-Verlag*, 1981.

[22] A. Abdulkadiroglu, P. A. Pathak, and A. E. Roth, “The boston public school match,” *American Economic Review*, vol. 95, no. 2, pp. 368—-371, 2005.

[23] A. Abdulkadiroglu, P. A. Pathak, and A. E. Roth, “The new york city high school match,” *American Economic Review*, vol. 95, no. 2, pp. 364—367, 2005.

[24] A. Abdulkadiroglu, P. A. Pathak, A. E. Roth, and T. Sonmez, “Changing the boston school-choice mechanism,” *NBER working paper 11965*, 2006.

[25] M. Balinski and T. Sonmez, “A tale of two mechanisms: student placement,” *Journal of Economic Theory*, vol. 84, no. 1, pp. 73—-94, 1999.

[26] S. Braun, N. Dwenger, and D. Kubler, “Telling the truth may not pay off: An empirical study of centralized university admissions in germany,” *The B.E. Journal of Economic Analysis and Policy*, vol. 10, no. 1, pp. 73—-94, 2010.

[27] A. Romero-Medina, “Implementation of stable solutions in a restricted matching market,” *Review of Economic Design*, vol. 3, no. 2, pp. 137—147, 2010.

[28] Y. Chen and T. Sonmez, “Improving efficiency of on-campus housing: An experimental study,” *American Economic Review*, vol. 92, no. 5, pp. 1669—-1686, 2002.

[29] N. Perach, J. Polak, and U. G. Rothblum, “A stable matching model with an entrance criterion applied to the assignment of students to dormitories at the technion,” *International Journal of Game Theory*, vol. 36, no. 3–4, pp. 519—-535, 2008.

[30] N. Garg, T. Kavitha, A. Kumar, K. Mehlhorn, and J. Mestre, “Assigning papers to referees,” *Algorithmica*, vol. 58, no. 1, pp. 119—-136, 2010.

[31] E. Kujansuu, T. Lindberg, and E. Makinen, “The stable roommates problem and chess tournament pairings,” *Divulgaciones Matematicas*, vol. 7, no. 1, pp. 19–28, 1999.

[32] D. J. Abraham, A. Blum, and T. Sandholm, “Clearing algorithms for barter exchange markets: enabling nationwide kidney exchanges,” *Proceedings of EC ’07: the 8th ACM Conference on Electronic Commerce(ACM)*, pp. 295—-304, 2007.

[33] A. E. Roth, T. Sonmez, and M. U. Unver, “Efficient kidney exchange:coincidence of wants in a market with compatibility-based preferences,” *American Economic Review*, vol. 97, no. 3, pp. 828—-851, 2007.

[34] D. F. Manlove and G. O’Malley, “Paired and altruistic kidney donation in the uk: Algorithms and experimentation,” *Proceedings of SEA ’12: the 11th International Symposium on Experimental Algorithms, Lecture Notes in Computer Science*, vol. 7276, pp. 271—282, 2012.

[35] A. T. Gai, F. Mathieu, F. de Montgolfier, and J. Reynier, “Stratification in p2p networks: Application to bittorrent,” *Proceedings of ICDCS ’07: the 27th IEEE International Conference on Distributed Computing Systems (IEEE Computer Society)*, 2007.

[36] D. Lebedev, F. Mathieu, L. Viennot, A. T. Gai, J. Reynier, and F. de Montgolfier, “On using matching theory to understand p2p network design,” *Proceedings of INOC ’07: International Network Optimization Conference*, 2007.

[37] F. Mathieu, “Self-stabilization in preference-based systems,” *Peer-to-Peer Networking and Applications*, vol. 1, no. 2, pp. 104—-121, 2008.

[38] F. Mathieu, “Acyclic preference-based systems: A self-stabilizing approach for p2p systems,” *X. Shen, H. Yu, J. Buford and M. Akon (eds.), Handbook of Peer-to-Peer Networking (Springer)*, pp. 1165—-1203, 2010.

[39] A. E. Roth, T. Sonmez, and M. U. Unver, “Pairwise kidney exchange,” *Journal of Economic Theory*, vol. 125, pp. 151—-188, 2005.

[40] A. E. Roth, “The economics of matching: Stability and incentives,” *Mathematics of Operations Research*, vol. 7, no. 4, pp. 617—-628, 1982.

[41] L. E. Dubins and D. A. Freedman, “Machiavelli and the gale-shapley algorithm,” *American Mathematical Monthly*, vol. 88, no. 7, pp. 485—-494, 1981.

[42] G. Demange, D. Gale, and M. Sotomayor, “A further remark on the stable matching problem,” *Discrete Applied Mathematics*, vol. 16, pp. 217—-222, 1987.

[43] D. Gale and M. Sotomayor, “Ms. machiavelli and the stable matching problem,” *American Mathematical Monthly*, vol. 92, no. 4, pp. 261—-268, 1985.

[44] C. P. Teo, J. Sethuraman, and W. P. Tan, “Gale-shapley stable marriage problem revisited: strategic issues and applications,” *Management Science*, vol. 47, no. 9, pp. 1252—-1267, 1999.

[45] “Canadian resident matching service website.” http://www.carms.ca, September 2020.

[46] “Scottish foundation allocation scheme website.” http://www.scotmt.scot.nhs.uk/foundation/foundation-programmes/.aspx, September 2020.

[47] H. Zhang, “An analysis of the chinese college admission system,” *Ph.D. thesis, University of Edinburgh, School of Economics*, 2009.

[48] H. W. Kuhn, “The hungarian method for the assignment problem,” *Naval Research Logistics Quarterly*, vol. 2, pp. 83–97, 1955.

[49] H. W. Kuhn, “Variants of the hungarian method for assignment problems,” *Naval Research Logistics Quarterly*, vol. 3, pp. 253–258, 1956.

[50] J. Munkres, “Algorithms for the assignment and transportation problems,” *Journal of the Society for Industrial and Applied Mathematics*, vol. 5, no. 1, pp. 32–38, 1956.

[51] C. Y. Teo and D. J. Ho, “A systematic approach to the implementation of final year project in an electrical engineering undergraduate course,” *IEEE Transactions on Education*, vol. 41, no. 1, 199.

[52] L. G. Proll, “A simple method of assigning projects to students,” *Operational Research Quarterly*, vol. 23, no. 2, pp. 195—-201, 1972.

[53] J. Dye, “A constraint logic programming approach to the stable marriage problem and its application to student-project allocation,” *BSc Honours project report, University of York, Department of Computer Science*, 2001.

[54] D. Kazakov, “Co-ordination of student-project allocation,” *Manuscript, University of York, Department of Computer Science*, 2002.

[55] M. Thorn, “A constraint programming approach to the studentproject allocation problem,” *BSc Honours project report, University of York, Department of Computer Science*, 2003.

[56] K. Iwama, S. Miyazaki, and H. Yanagisawa, “Improved approximation bounds for the student-project allocation problem with preferences over projects,” *Journal of Discrete Algorithms*, vol. 13, pp. 59—66, 2012.

[57] D. F. Manlove and G. O’Malley, “Student-project allocation with preferences over projects,” *Journal of Discrete Algorithms*, vol. 6, no. 4, pp. 553—560, 2008.

[58] P. R. Harper, V. d. Senna, I. T. Vieira, and A. K. Shahani, “A genetic algorithm for the project assignment problem,” *Comput. Oper. Res.*, vol. 32, no. 5, pp. 1255—1265, 2005.

[59] J. E. Beasley, “Or-library: distributing test problems by electronic mail,” *J.Oper. Res. Soc.*, vol. 41, no. 11, pp. 1069—1072, 1990.

[60] V. Sanchez-Anguix, R. Chalumuric, R. Aydogan, and V. Julian, “A near pareto optimal approach to student-supervisor allocation with two sided preferences and workload balance,” vol. 38, no. 5, pp. 593–614, 1997.

[61] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: Nsga-ii,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182—197, 2002.

[62] C. A. C. Coello, G. B. Lamont, and D. A. V. Veldhuizen, “Evolutionary algorithms for solving multi-objective problems,” *Springer, New York*, vol. 2nd Ed., 2007.

[63] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization:a short review," *2008 IEEE Congress on Evolutionary Computation CEC 2008 (IEEE World Congress on Computational Intelligence)*, pp. 2424—2431, 2008.

[64] T. L. Saaty, "Decision making with the analytic hierarchy process," *Int. J. Services Sciences*, vol. 1, no. 1, 2008.

[65] J. Figuera, S. Greco, and M. Ehrgott, "Multiple criteria decision analysis, state of the art surveys," *Springer*, 2005.

[66] R. Irving, "Matching medical students to pairs of hospitals: a new variation on a well-known theme," *Proceedings of ESA '98: the 6th European Symposium on Algorithms, Lecture Notes in Computer Science*, vol. 1461, pp. 381—-392, 1998.

[67] P. Biro, "Student admissions in hungary as gale and shapley envisaged," *Tech. Rep. TR-2008-291, University of Glasgow, Department of Computing Science*, 2008.

[68] P. Biro and S. Kiselgof, "College admissions with stable score-limits," *Proceedings of COMSOC '12: the 4th International Workshop on Computational Social Choice*, 2012.

[69] F. Mathieu, "Upper bounds for stabilization in acyclic preferencebased systems," *Proceedings of SSS '07: the 9th International Symposium on Stabilization, Safety, and Security of Distributed Systems, Lecture Notes in Computer Science*, vol. 4838, pp. 372—-382, 2007.

[70] V. Lai, B. K. Wong, and W. Cheung, "Group decision making in a multiple criteria environment: A case using the ahp in the software selection," *European Journal of Operational Research*, vol. 137, no. 1, pp. 134—-144, 2002.

[71] K. M. A. Harbi, "Application of ahp in project management," *International Journal of Project Management*, vol. 19, no. 4, pp. 19—-27, 2001.

[72] R. J. Kuo, S. C. Chi, and S. S. Kao, "A decision support system for locating convenience store through fuzzy ahp," *Computers and Industrial Engineering*, vol. 37, no. 1–2, pp. 323—326, 1999.

[73] H. W. Jung and B. Choi, "Optimization models for quality and cost of modular software systems," *European Journal of Operational Research*, vol. 112, no. 3, pp. 613—-619, 1999.

[74] G. A. Forgionne and R. Kohli, "A multi-criteria assessment of decision technology system and journal quality," *Information and Management*, vol. 38, no. 7, pp. 421—435, 2001.

[75] M. Badri, "Combining the ahp and gp for global facility location–allocation problem," *International Journal of Production Economics*, vol. 62, no. 3, pp. 237—-248, 1999.

[76] C. W. Lee and N. K. Kwak, "Information resource planning for a health-care system using an ahp-based goal programming method," *Journal of Operational Research Society*, vol. 50, pp. 1191—1198, 1999.

[77] T. Padma and P. Balasubramanie, "Knowledge based decision support system to assist work-related risk analysis in musculoskeletal disorder," *Knowledge-Based Systems*, vol. 22, no. 1, pp. 72—78, 1995.

[78] I. Miyaji, Y. Nakagawa, and K. Ohno, "Decision support system for the composition of the examination problem," *European Journal of Operational Research*, vol. 80, no. 1, pp. 130—138, 1995.

[79] S. J. Chen and C. L. Hwang, "Fuzzy multiple attribute decision making: Methods and applications," *Springer-Verlag*, 1992.

[80] K. Yoon and C. L. Hwang, "Multiple attribute decision making," *Thousand Oaks, CA: Sage Publication*, 1992.

[81] E. K. Aydogan, "Performance measurement model for turkish aviation firms using the rough-ahp and topsis methods under fuzzy environment," *Expert Systems with Applications*, vol. 38, pp. 3992—3998, 2011.

[82] A. Kelemenis, K. Ergazakis, and D. Askounis, "Support managers' selection using an extension of fuzzy topsis," *Expert Systems with Applications*, vol. 38, pp. 2774—-2782, 2011.

[83] F. E. Boran, S. Genç, M. Kurt, and D. Akay, "Personnel selection based on intuitionistic fuzzy sets," *Human Factors and Ergonomics in Manufacturing Service Industries*, vol. 38, pp. 1—-11, 2011.

[84] S. Mahmoodzadeh, J. Shahrabi, M. Pariazar, and M. S. Zaeri, "Project selection by using fuzzy ahp and topsis technique," *World Academy of Science, Engineering and Technology International Journal of Industrial and Manufacturing Engineering*, vol. 1, no. 6, 2007.

[85] M. F. Chen and G. H. Tzeng, "Combining gray relation and topsis concepts for selecting an expatriate host country," *Mathematical and Computer Modelling*, vol. 40, pp. 1473–1490, 2004.

[86] M. Janic, "Multicriteria evaluation of high-speed rail, transrapid maglev, and air passenger transport in europe," *Transportation Planning and Technology*, vol. 26, no. 6, pp. 491—512, 2003.

[87] C. K. Kwong and S. M. Tam, "Case-based reasoning approach to concurrent design of low power transformers," *Journal of Materials Processing Technology*, vol. 128, pp. 136—-141, 2002.

[88] A. S. Milani, A. Shanian, and R. Madoliat, "The effect of normalization norms in multiple attribute decision making models: A case study in gear material selection," *Structural Multidisciplinary Optimization*, vol. 29, no. 4, pp. 312–318, 2005.

[89] B. Srdjevic, Y. D. P. Medeiros, and A. S. Faria, "An objective multi-criteria evaluation of water management scenarios," *Water Resources Management*, vol. 18, pp. 35–54, 2004.

[90] T. Yang and P. Chou, "Solving a multiresponse simulation–optimization problem with discrete variables using a multi-attribute decision-making method," *Mathematics and Computers in Simulation*, vol. 68, pp. 9–21, 2005.

[91] K. Yoon and C. L. Hwang, "Manufacturing plant location analysis by multiple attribute decision making: Part i—single-plant strategy," *International Journal of Production Research*, vol. 23, pp. 345–359, 1985.

[92] Y. Lai, "Topsis for modm," *European Journal of Operational Research*, vol. 76, pp. 486–500, 1994.

[93] H. S. Shih, W. Y. Lin, and E. S. Lee, "Group decision making for topsis," *Joint 9th IFSA World Congress and 20th NAFIPS International Conference, IFSA/NAFIPS 2001*, pp. 2712—-2717, 2001.

[94] G. Brown and M. Atkins, "Effective teaching in higher education," *London: Methuen*, 1988.

[95] M. Pearson and C. Kayrooz, "Enabling critical reflection on research supervisory practice," *International Journal for Academic Development*, vol. 9, no. 1, pp. 99–116, 2004.

[96] M. L. Orellana, A. Darder, A. Pérez, and J. Salinas, "Improving doctoral success by matching phd students with supervisors," *International Journal of Doctoral Studies*, vol. 11, pp. 87–103, 2016.

[97] H. B. Rashid, A. T. Priyoti, and S. Sharmin, "Undergraduate cse thesis supervisor selection in context of bangladesh using stable matching," *Undergraduate Thesis, Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology*, 2019.

[98] N. R. Kuncel, S. A. Hezlett, and D. S. Ones, "A comprehensive meta-analysis of the predictive validity of the graduate record examinations: Implications for graduate student selection and performance," *Psychological Bulletin*, vol. 127, pp. 162–181, 2001.

[99] N. R. Kuncel and S. A. Hezlett, "Standardized tests predict graduate students' success," *Science*, vol. 315, p. 1080–1081, 2007.

[100] P. L. Craig, J. J. Gordon, R. M. Clark, and V. Langendyk, "Prior academic background and student performance in assessment in a graduate entry programme," *Medical Education*, vol. 38, no. 11, pp. 1164–1168, 2004.

[101] S. J. Cubillo, J.M. and J. Cerviňo, "International students' decision-making process," *International Journal of Educational Management*, vol. 20, no. 2, pp. 101–115, 2006.

[102] J. Ivy, "Choosing futures: Influence of ethnic origin in university choice," *International Journal of Educational Management*, vol. 24, no. 5, pp. 391–403, 2010.

[103] T. Johnston, "Who and what influences choice of university? student and university perceptions," *American Journal of Business Education*, vol. 3, no. 10, pp. 15–23, 2010.

[104] K. Tucciarone, "Advertising's effect on community college search and choice," *The Community College Enterprise*, vol. Fall, pp. 73–91, 2008.

[105] G. D. Malaney, "Why students pursue graduate education, how they find out about a program, and why they apply to a specific school," *College and University*, vol. 62, no. 3, pp. 247–258, 1987.

[106] C. Olson and M. A. King, "A preliminary analysis of the decision process of graduate students in college choice," *College and University*, vol. 60, no. 4, pp. 304–315, 1985.

[107] R. E. Kallio, "Factors influencing the college choice decisions of graduate students," *Research in Higher Education*, vol. 36, pp. 109–124, 1995.

[108] M. C. Poock and P. G. Love, "Factors influencing the program choice of doctoral students in higher education administration," *NASPA Journal*, vol. 38, no. 2, pp. 203–223, 2001.

[109] M. Saunders, P. Lewis, and A. Thornhill, "Research methods for business students," *Pearson Education Limited*, vol. 6, 2012.

[110] S. M. Belenson and K. C. Kapur, "An algorithm for solving multicriterion linear programming problems with examples," *Operational Research Quarterly*, vol. 24, no. 1, pp. 65––77, 1973.

[111] M. Zeleny, "A concept of compromise solutions and the method of the displaced ideal," *Computers and Operations Research*, vol. 1, pp. 479—-496, 1974.

[112] P. Kenekayoro, "Student project allocation dataset." https://figshare.com/articles/Student_Project_Allocation_Dataset/6490451, September 2020.

[113] E. Zitzler and S. Kunzli, "Indicator-based selection in multiobjective search," *Springer*, vol. 3242, pp. 832—-842, 2004.

[114] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated pareto sets, moea/d and nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 284–302, 2009.