UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

SCHOLARSHIP-STUDENT MATCHING PROCESS OPTIMIZATION

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE

By

SEKAR RIZKY RACHMAWATI
Norman, Oklahoma
2017

SCHOLARSHIP-STUDENT MATCHING PROCESS OPTIMIZATION

A THESIS APPROVED FOR THE
GALLOGLY COLLEGE OF ENGINEERING

BY

Dr. Randa L. Shehab, Chair

Dr. Charles Nicholson

Dr. Christan Grant

To my daughters, Primrose and Linnea,

whom I aspire myself to be a woman they would look up to.

To my parents,

who support me in whichever path I choose to pursue.

To my husband,

my hero among heroes.

Acknowledgements

It does take a village to write a thesis. I would like to thank and acknowledge all the parties involved on this rollercoaster-like journey. My advisor, Dr. Randa Shehab, I would like to thank her for all the support and feedback she gave me along this journey. She is not only my advisor, she becomes a woman that I would look up to in my future. I would also like to thank my thesis committee members, Dr. Charles Nicholson, for his feedback on this work, and Dr. Christan Grant, for introducing me to Python.

This journey will not be possible without the support from the Fulbright Program. I would like to thank them for the opportunity they gave me to study at the University of Oklahoma.

To Nicola Manos, the program director of Data Science and Analytics, I would like to thank for all the encouragement she gave me along this journey. We, DSA students, are very lucky to have her as our program director. Jackie Foos, I would like to thank her for all the information and discussion related to this work. Jessica Orr, I would like to thank her for helping me to proofread this thesis. My best friends, Jiwon Jeon, Alexander David, and Yunjie Wen. I am grateful for all their help and support during this journey. They helped to make this rollercoaster ride bearable. Also, the Dalimunte family, I would like to thank them for giving me a place to call "home".

To my parents and my sister, I would like to express my deepest gratitude for taking care of my daughters while I have been far away from home. My husband, Miswar, for being (virtually) with me every step of the way and for holding down the fort while I pursued my dream, thank you. And for Primrose and Linnea, Mamak will come back, soon.

Table of Contents

List of Tables

List of Figures

Abstract

Students need the support of financial aid to help them pay for college tuition and fees. One source of financial aid comes from scholarships. Universities are a primary source for scholarships. The University of Oklahoma Gallogly College of Engineering awards a number of scholarships to students each academic year. The committee who is responsible for the distribution of scholarships has to decide which student applicant receives a particular scholarship. This thesis focuses on how to optimize the matching of scholarships and students, taking into consideration the requirements of the scholarship and the credentials of students who are applying for the scholarship. This thesis approaches the process of matching students with scholarships in two ways. First, matching can be done from the scholarship side, which only considers the requirements of the scholarship and ranks the students based on how well they meet the scholarship requirements. Second, matching can be done from both sides, considering the requirements of the scholarship and the value of the award for the student. The matching results show that the first approach has a higher number of matching sets of scholarship and student.

Chapter 1: Introduction

The cost of attending higher education in the United States is increasing every year (Snyder, de Brey, & Dillow, 2016). Students need the support of financial aid to help them pay for college tuition and fees (Snyder et al., 2016). One source of financial aid comes from scholarships. Scholarships can be merit-based or need-based. Universities are a primary source for scholarships, often through gifts from alumni and friends.

The University of Oklahoma Gallogly College of Engineering (GCoE) awards a number of scholarships to students each academic year. The GCoE manages the distribution of scholarships for engineering students. The committee who is responsible for the distribution of scholarships has to decide which student applicant receives a particular scholarship. Given that each scholarship has specific criteria, defined at the discretion of the donor, the matching of students to scholarships is not easily accomplished. Currently, the process is done manually and can take the team two to three days to accomplish.

This thesis focuses on how to optimize the matching of scholarships and students, taking into consideration the requirements of the scholarship and the credentials of students who are applying for the scholarship. The process of matching students with scholarships can be approached in two ways with different objectives in mind. First, matching can be done from the scholarship side, which only considers the requirements of the scholarship and ranks the students based on how well they meet the scholarship requirements. The objective of this approach is to maximize the number of scholarships awarded to students. Second, matching can be done from both sides, considering the requirements of the scholarship and the value of the award for the student. The objective of this approach is to maximize the value of the scholarship award for the student given that the student meets the scholarship requirements.

This thesis is structured as follows. Chapter 2 contains background on the concepts related to this work, including bipartite graph, network flow, and stable matching. In Chapter 3, the methodology for the work is presented, which includes the methods to prepare the dataset and the approaches for the matching process. Chapter 4 describes the results and the evaluation of the results. The conclusion of the work is presented in Chapter 5.

Chapter 2: Background

A matching problem involves a set of participants, in which each participant has a certain capacity and a subset of the participants have preference over other participants. The definition of matching in this context is the attempt to assign each participant to one or more qualified participant(s) based on the preferences of the participants, without exceeding the capacity of the participants (Sng, 2008).

There are several problems in the real-world which can be classified as a matching problem. A popular one is assigning graduate medical students to hospital posts (Irving, 1998; Roth & Peranson, 1999). In the United States, it is known as National Resident Matching Program (NRMP). Another matching problem can be found in the implementation of assigning schools to students (Abdulkadiroglu & Sönmez, 2003; Aksoy et al., 2013). Two cities in the United States which applied such a centralized matching system are New York (Abdulkadiroğlu, Pathak, & Roth, 2005) and Boston (Abdulkadiroğlu, Pathak, Roth, & Sönmez, 2005). Another application of the matching concept is used for the system in managing the kidney exchange (Roth, Sönmez, & Ünver, 2004). The matching concept is also used to allocate students to courses (Diebold, Aziz, Bichler, Matthes, & Schneider, 2014) and projects (Abraham, Irving, & Manlove, 2003; Manlove & O'Malley, 2008)

There are several ways to classify the matching problem. The most distinct one would be bipartite and non-bipartite matching. A bipartite matching is when the participants can be divided into two disjoint sets (Sng, 2008), while in a non-bipartite model the participants are a single set. Several applications mentioned above (e.g., assigning graduate medical students to hospital posts, assigning schools to students, and allocating students to course) are bipartite matching, whereas the example for the non-bipartite model can be found in the assigning roommates problem (Irving, 1985). Furthermore, the matching problem can be divided based on the types of preferences list which are involved (one-sided or two-sided) and the types of mapping to assign the members from one side to the other (one-to-one and one-to-many).

The matching between scholarship and students in this thesis can be considered as a bipartite matching (it involves two disjoint set, scholarship, and student) with one-to-many mapping (each scholarship has a capacity for a number of students). We approach the matching between scholarships and students with one-sided preference (scholarship) and two-sided preference (scholarship and student).

This chapter presents the background concept for the matching problem. Section 2.1 describes the bipartite matching in graph theory, section 2.2 describes the flow in a network, and section 2.3 described the Stable Marriage Problem and Hospital-Resident problem.

2.1 Matching in a Graph

A graph G (V, E) is a pair of sets, consisting of a set of vertices $V = \{v_1, v_2, v_3, \ldots, v_n\}$ and a set of edges $E = \{e_1, e_2, e_3, \ldots, ep\}$; each edge has two endpoints which are members of V (Diestel, 2005; Even, 2011). Any two nodes connected by an edge are said to be adjacent. A bipartite graph is a graph where its vertices can be separated into two disjoint sets, and the vertices in the same class cannot be adjacent (Diestel, 2005). The bipartite graph can be represented as graph G (X, Y, E) with $X = \{x_1, x_2, x_3, \ldots, x_n\}$, $Y = \{y_1, y_2, y_3, \ldots, y_m\}$, and $E = \{e_1, e_2, e_3, \ldots, e_p\}$ which $V = X \cup Y$ is the set of vertices and E is the set of edges which each edge has one vertex in X and one in Y (Tanimoto, Itai, & Rodeh, 1978). Figure 1 shows an example of a bipartite graph.

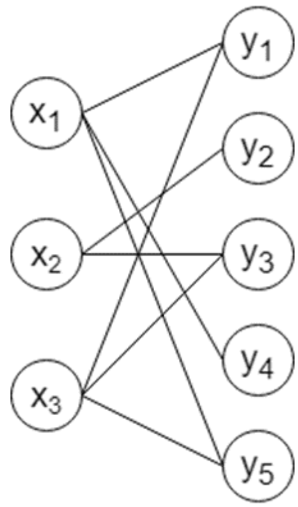

Figure 1: An example of bipartite graph

Matching in a bipartite graph G is a set of edges, M, where M is a subset of E such that no two edges in M share common vertices. An edge e = (x, y) ∈ E, where x ∈ X and y ∈ Y, is matched when e ∈ M, otherwise e is unmatched.

The size or cardinality of the matching, denoted by |M|, is the number of edges in M. A matching M is said to be *maximal* when M is not a proper subset of any other matching in G. A matching M is said to be *maximum* when M has the largest number of edges. A maximum matching is a maximal one, but not always the other way around. We can use a network flow technique to find a maximum matching in a bipartite graph (Even, 2011).

2.2 Network Flow

A directed graph N (V, E) is a network if V contains a source vertex, s, and sink vertex, t (where indegree(s) = outdegree(t) = 0); every edge e = (x, y) has non-negative capacity u (x, y) ≥ 0; and every vertex lies on the path between s and t (Abraham, 2003).

A flow function in N is an assignment of a real number f(e) to each edge e with the following conditions (Even, 2011):

- For every edge $e \in E$, $0 \le f(e) \le u(e)$
- For every vertex $x \in V$ except {s, t}, the input flow of x is equal to the output flow of x.

The size of the flow is the total flow, which is the net sum of flow into the sink (Even, 2011). Given a flow network N, the maximum flow problem is to find the maximum size of flow in N (Ahuja, Magnanti, & Orlin, 1993). Ahuja et al. (1993) describe several algorithms to solve the maximum flow problem.

The bipartite matching can be modeled as network flow (Even, 2011). Let G (X, Y, E) be a bipartite graph shown in Figure 1, where each vertex x has capacity $b(x) \ge 1$. We can find the maximum matching for the bipartite graph in Figure 1 by turning the bipartite graph into a network flow. The network flow N is constructed with one vertex for each $x \in X$ and one vertex for each $y \in Y$. An edge (x, y) with capacity one is added whenever $(x, y) \in G$. Edges (s, x) from s to each vertex x with capacity b(x), and edges (y, t) for each vertex y to sink t with capacity one are also added. Figure 2 shows the network flow for the bipartite graph in Figure 1. Let f be the maximum flow of N, f is the maximum matching for a bipartite graph G (Even, 2011).

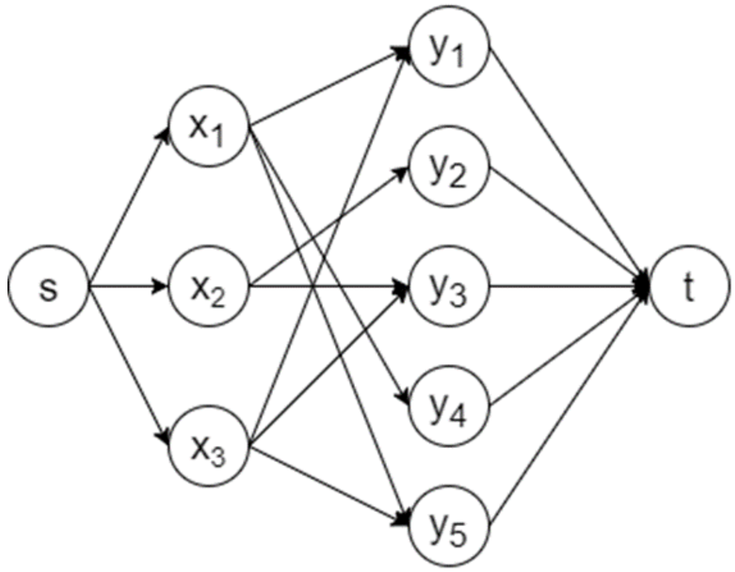

Figure 2: Network flow for bipartite graph in Figure 1

Let each edge e ∈ N be associated with cost c, where c(e) ≥ 0. The cost of the flow is the sum of the cost of the flow in each edge. Ahuja et al. (1993) describe several approaches to find the maximum flow minimum cost of N.

2.3 Stable Matching

A stable matching problem consists of a set of participants, each of whom has preference list ranking over a subset of other participants which they want to be paired up with. The problem is to produce a matching M of the participants such that no two participants prefer each other to their assignment in M.

Stable marriage problem is a stable matching problem with one-to-one mapping, whereas the hospital-resident matching problem is a stable matching problem with one-to-many mapping (Gusfield & Irving, 1989).

2.3.1 Stable Marriage Problem

An instance I of the Stable Marriage problem (SM) consists of two disjoint sets, men, U, and women, W, with |U|= |W| = n. Each person p in U ∪ W has preference list over the person on the other set, each man has preference over all women and each woman has preference over all men (Iwama & Miyazaki, 2008). This preference list is strictly ordered.

An assignment in M is a subset of U×W such that (m, w) ∈ M only if m and w find each other acceptable. If (m, w) ∈ M, we say that m is assigned to w and w is assigned to m. A matching is an assignment M such that each man is assigned to at most one woman in M, and each woman is assigned to at most one man in M. If (m, w) ∈ M, we say that m is matched to w and w is matched to m. We can denote w as M(m) and m as M(w). A blocking pair is a pair of man and woman (m, w) where m prefers w to M(m) and w prefers m to M(w). A matching is said to be stable if it admits no blocking pairs (Gusfield & Irving, 1989).

SM can be solved with the deferred acceptance algorithm, widely known as the Gale-Shapley Algorithm (Gale & Shapley, 1962). The algorithm involves several iterations of "proposal" from one set (men) to the other set (women) or the other way around, from women to men. Figure 3 (Gusfield & Irving, 1989) shows the basic Gale-Shapley algorithm when the men are proposing.

```
Algorithm: Deferred Acceptance (Gale-Shapley)

assign each person to be free;
while some man m is free do
begin
        w := first woman on m's list to whom m has not yet proposed;
        if w is free then
                assign m and w to be engaged {to each other}
        else
                if w prefers m to her fiance m' then
                        assign m and w to be engaged and m' to be free
                else
                        w rejects m {and m remains free}
end;
output the stable matching consisting of the n engaged pairs
```

Figure 3: Basic Gale-Shapley Algorithm (Gusfield & Irving, 1989)

The algorithm always finds a stable matching for an instance of SM (Gale & Shapley, 1962). If the men are the proposer, the algorithm is known as man-oriented. Otherwise, it is known as woman-oriented. The algorithm involves nondeterminism because the order in which the proposer proposes is of no consequence to the result (Gusfield & Irving, 1989).

The man-oriented algorithm gives the man-optimal matching result where each man has the best partner that he can have in any stable matching, while the woman-oriented algorithm gives the woman-optimal matching result where each woman gets the best partner she can have in any stable matching (Gusfield & Irving, 1989). The man-optimal is also woman-pessimal because each woman gets the worst partner she can have on any stable matching (McVitie & Wilson, 1971), whereas the woman-optimal is the man-pessimal.

There are several variations of Stable Marriage Problem. These variations relate to the conditions of the preference list.

2.3.1.1 Incomplete List

SMI (Stable Marriage with Incomplete List) is a variant of the SM where each person need not include all the members of the opposite set in the preference list (Iwama & Miyazaki, 2008). The preference list of person p only includes the member of the opposite set which person p finds acceptable. A person p considers a person q acceptable if and only if q is on the preference list of p.

A man m and woman w are assigned to each other in a matching M only if m and w are acceptable to one another. Thus, the matching need not be complete, because not all the members of either set need to be assigned. A blocking pair is a pair of man and woman (m, w) where:

- m and w find each other acceptable
- either m is unassigned in M, or m prefers w to M(m)
- either w is unassigned in M, or w prefers m to M(w)

A matching in SMI is said to be stable if it admits no blocking pair. Every instance of SMI admits a stable matching (Gale & Shapley, 1962). The extended version of the Gale-Shapley algorithm can be used to find a stable matching in any instances of SMI (Gusfield & Irving, 1989). For any matching M in an instance of SMI, some persons may be unassigned in M, but the same persons are unassigned in all stable matching. Therefore, the cardinality of all stable matching for an instance of SMI is the same (Gale & Sotomayor, 1985).

2.3.1.2 Ties

SMT is a variant of the SM where the preference list of each person includes all the members of the opposite set but can contain ties (i.e., several persons can have the same rank) (Iwama & Miyazaki, 2008). A person p is said to be indifferent to persons q and r if q and r appear in a tie in the preference list of p. The existence of ties introduces three definitions of stability for a matching, namely weakly, strong, and super stability (Irving, 1994).

A matching M is weakly stable if there is not any blocking pair (m, w) where m and w prefer each other to their assigned partner in M. A weakly stable matching can always be found for any instance of SMT by breaking ties arbitrarily and applying the Gale-Shapley algorithm. This method produces weakly stable matching for any instance of SMT (Gusfield & Irving, 1989).

A matching M is strongly stable if it admits no blocking pair (m, w) such that either:

- m prefers w to M(m), and either w prefers m to M(w) or is indifferent between them
- w prefers m to M(w) and either m prefers w to M(m) or is indifferent between them.

Strongly stable matching need not exist for a given instance of SMT. There is an algorithm to check whether a given instance has a strongly stable matching, and to find one if one exists (Irving, 1994).

A matching M is super stable if it admits no blocking pair (m, w) such that:

- m either prefers w to M(m) or is indifferent between them, or
- w either prefers m to M(w) or is indifferent between them.

Super stable matching need not exist for a given instance of SMT. There is an algorithm to check whether a given instance has a super stable matching, and to find one if there exist (Irving, 1994).

2.3.1.3 Incomplete List and Ties

The variant of stable matching with incomplete list and ties is the combination of the two previous variants; we denote this problem as SMTI (Stable Marriage with ties and incomplete list). SMTI has an incomplete preference list which can contain ties (Iwama & Miyazaki, 2008). The notion of stability in SMTI consists of weakly, strong, and super stable. A weakly stable matching can be found with the same method as in SMT. The result of weakly stable matching can have different cardinality. The problem to find the maximum cardinality for weakly stable matching in SMTI is NP-Hard (Manlove, Irving, Iwama, Miyazaki, & Morita, 2002).

2.3.2 Hospital-Resident Matching

An instance I of the Hospital-Resident problem (HR) consists of two disjoint sets of hospitals H and residents R. Each resident r has a preference list which ranks a subset of H in strict order. Each hospital h has a preference list which ranks in strict order the residents who ranked h in their preference list. We say r and h are acceptable to each other if they rank each other on their preference list. Each hospital h has a capacity of c which is the maximum number of residents that can be assigned to h (Gusfield & Irving, 1989; Iwama & Miyazaki, 2008).

An assignment M is a subset of $H \times R$ such that $(h, r) \in M$, implies that h and r find each other acceptable. If $(h, r) \in M$, we can say that h is assigned to r and r is assigned to h. For each hospital $h \in H$, $M(h)$ denotes the set of residents assigned to h in

M. h is said to be fully subscribed in M if $|M(h)|=c$ and undersubscribed in M if $|M(h)|<c$. A matching M is an assignment where each resident is assigned to at most one hospital, and each hospital is assigned to at most c residents.

A matching M is stable if it admits no blocking pair (h, r) such that:

- h and r find each other acceptable,
- either r is unassigned in M or r prefers h to M(r), and
- either h is undersubscribed, or h prefers r to the worst assigned resident in M(h)

Every instance of HR admits a stable matching (Gusfield & Irving, 1989). The algorithm for SM can be extended for HR. The definition of man-optimal and woman-optimal can be extended to hospital-optimal and resident-optimal respectively. Figure 4 shows the algorithm for the hospital-oriented, while Figure 5 shows the algorithm for resident-oriented (Gusfield & Irving, 1989).

```
Algorithm: Hospital-oriented

assign each resident to be free;
assign each hospital to be totally unsubscribed;
while (some hospital h is undersubscribed) and
        (h's list contains a resident r not provisionally assigned to h) do
begin
        r := first such resident on h's list;
        if r is already assigned, say to h', then
                break the provisional assignment of r to h';
        provisionally assign r to h;
        for each successor h' of h on r's list do
                remove h' and r from each other's lists
end
```

Figure 4: Hospital-oriented algorithm (Gusfield & Irving, 1989)

```
Algorithm: Resident-oriented

assign all residents to be free;
assign all hospitals to be totally unsubscribed;
while (some resident r is free) and (r has nonempty list) do
begin
        h := first hospital on r's list; {r "proposes" to h}
        if h is fully subscribed then
                begin
                        r' := worst resident provisionally assigned to h;
                        assign r' to be free
                end;
        provisionally assign r to h;
        if h is fully subscribed then
                begin
                        s := worst resident provisionally assigned to h;
                        for each successor s' of s on h's list do
                                remove s' and h from each other's lists
                end
end
```

Figure 5: Resident-oriented algorithm (Gusfield & Irving, 1989)

A Hospital-Resident problem with ties (HRT) is the variant of HR where the preference of hospital and resident may contain ties. The definition of stability is similar to the definition of stability for SMTI. To find a weakly stable matching in an instance of HRT, the method to find a weakly stable matching in SMTI can be applied. Similar to the SMTI, the problem of finding a maximum cardinality stable matching in HRT is NP-Hard.

Chapter 3: Methodology

This chapter presents the methodology. I describe data preparation in section 3.1, the matching process in section 3.2, and the method to optimize the matching in section 3.3.

3.1 Data Preparation

The data for this research consists of scholarship criteria data and student attribute data. The data need to undergo preparation before the matching process. The data preparation consists of two independent processes, scholarship data preparation and student data extraction. The results are exported into a database at the end of each process. The database management system used in this research is MySQL.

3.1.1 Scholarship Data Preparation

Scholarship criteria data come from the agreement documents between donors and the Gallogly College of Engineering. The scholarship committee summarized the contents of the agreement documents into an Excel file. The scholarship data in the file consists of account number (identity of the scholarship), name, criteria (which must be satisfied by awardee characteristics), and preferences (preferred awardee characteristics) of the scholarships. Each scholarship has different criteria and preferences. We refer to the constraints and preferences as the scholarship requirements. These requirements are translated into attributes. Figure 6 represents the above description.

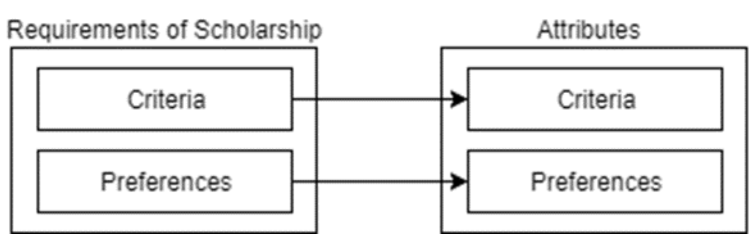

Figure 6: Mapping between requirements and attributes of scholarship

In addition to the agreement documents, the scholarship data has another source saved in an excel file. The data contain the total dollar amount available in each account. Through predetermined allocation amounts, the number of awards can be estimated. The committee also has a priority for the scholarships (which scholarship should be assigned first to students). At the end of this preparation process, the data for account, name, criteria, preferences, the total amount, the number of scholarship, and priority of the scholarship are stored in a database.

3.1.2 Student Data Extraction

Student data for the matching process come from a scholarship applicant report. The data are stored in an Excel file. The data consist of multiple attributes for each student. A process of extraction selects each student's identity attributes and several attributes which are of relevance to the attributes of scholarship. Figure 7 illustrates the description. The list of the attributes is in Appendix A.

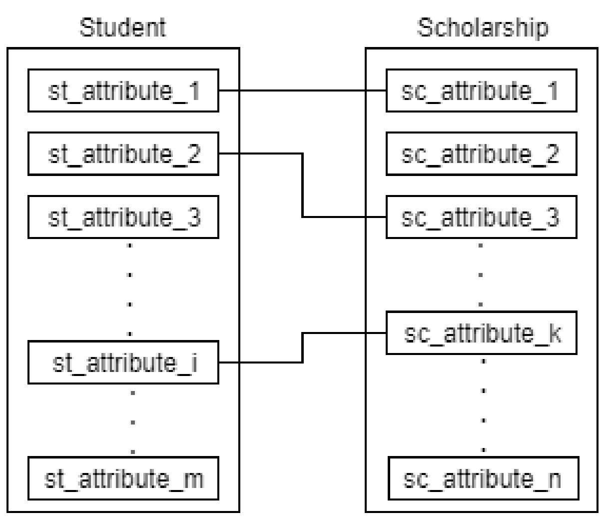

Figure 7: Mapping between student's attributes and scholarship's attributes

The process not only extracts student's attributes which are relevant to the scholarship's attributes, but also specific attributes of the students which are needed in the filtering process. The filtering process is a sub-process of the matching process which is explained in the next section. The result of the student data extraction process is exported into the same database as the scholarship data.

3.1.3 Dataset for Testing

The scholarship data for this research are a subset of the full scholarship data. After the exclusion of several unnecessary attributes, there are 18 attributes used for the matching process. The list of the attributes for this research is in Appendix A, while the list of excluded attributes is in Appendix B. There are 95 scholarships, with the total award of 333 awards, related to those attributes.

The student data for this research consist of 923 students. There are missing values in the student data. The missing values are caused by students who did not provide information. The missing values in the student data are treated as empty values (NULL) in the matching process. For the numeric attributes, such as Gross Financial Need and Grade Point Average, the missing values are considered as 0 value. Three different students datasets were generated from subsets of the student data and are used to test the matching process. Table 1 shows the different student datasets for testing.

Table 1: The student dataset for testing

Name of dataset	Number of students	Filter
full	923	0
sample 1	500	GPA > 2
sample 2	500	GPA > 2

The first dataset (full) is the full dataset of students with no filter. The second dataset (sample 1) contains 500 randomly selected students from the full dataset. The third dataset (sample 2) contains a different 500 randomly selected students from the full dataset. For the second and third datasets, students with GPAs less than or equal to two are filtered out in the matching process (the detail for the filtering process is in section 3.2.2).

3.2 Matching Process

The matching process consists of several sub-processes. The sub-processes are initial matching, filtering, ranking, and optimized matching. We can see the sequence of the process in Figure 8.

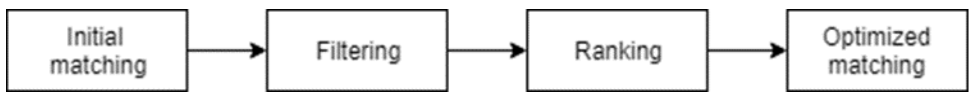

Figure 8: Sequence of the sub-processes in matching process

The scholarship-student matching process in this research uses several assumptions. Those assumptions are as follow:

1. One student is assigned to at most one scholarship.
2. The amount of award the student receives for a particular scholarship is the same for all students who receive that particular scholarship.
3. The following formula calculates the number of allocated award for a scholarship:

$$p_i = \left\lfloor \frac{\lfloor A_i \rfloor}{a_i} \right\rfloor$$

where a_i is the amount of award given to the student, $\lfloor A_i \rfloor$ is the total amount of award rounding down to the nearest thousands dollar value, and p_i is the number of award allocated to students for scholarship s_i

4. The students prefer scholarships which give a higher award (higher amount of dollar value) and those scholarships that give a higher number of awards.
5. The ranking process for student and scholarship is strictly ordered.
6. The committee has a set of priorities for the scholarships. The scholarships are assigned to students based on these priorities.

The matching process for an instance I of the scholarship-student matching involves a set of n scholarship $S = \{s_1, s_2, \ldots, s_n\}$ and a set of m students $T = \{t_1, t_2, \ldots, t_m)$ and is described in the following sections: 3.2.1 describes the initial matching, 3.2.2 describes the filtering, 3.2.3 describes the ranking, and 3.2.4 describes the optimized matching.

3.2.1 Initial Matching

After the data are stored in a database, we can begin the initial matching process. The initial matching selects subsets of students who qualify for each scholarship based on the criteria and preferences of the scholarship. Because each scholarship has different criteria and preferences, the subset of students who qualify can differ from one another. A student can qualify for several scholarships. Therefore, there are overlaps between the subsets of students for the scholarships.

For an instance I, with a set of scholarships S and a set of students T, the initial matching selects students who qualified for scholarship s_i. Student t_j is qualified for s_i if t_j fulfills all the criteria of s_i. The number of preferences of s_i which t_j fulfills is counted and used in the ranking process. D_i is the subset of students who qualify for scholarship s_i, and H_j is the subset of scholarships for which student t_j is qualified. Each scholarship has a capacity constraint p, with p_i being the number of allocated awards for scholarship s_i.

3.2.2 Filtering

This process removes several students from the set of students in the initial matching. There are two types of filters. The mass filter is the filter based on certain values of the students' attributes. This filter removes several students at once who have the defined value in the filter. The other filter, the individual filter, removes students based on an individual student's identity. There are several reasons for filtering out the students, e.g. the students are national merit scholars who are not eligible for additional awards, the students already received other scholarships, or the students have a missing value in their GPA. The filtering of individual students happens when the committee determines that a student is ineligible to receive a scholarship. The subsequent process, ranking, does not consider filtered students.

3.2.3 Ranking

To determine the final match between student and scholarship, the ranking process ranks the students matched to each scholarship and ranks the scholarships matched to each student. The students are ranked based on attributes of Gross Financial Need (GFN) and Grade Point Average (GPA), as requested by the scholarship committee. The students are ranked based on their GFN (primary rank), then their GPA (secondary rank). The scholarships for each student are ranked by the amount of award, the number of awards of the scholarship, and the priority of scholarship. The ranking process for the scholarships for each student is based on the assumptions that students prefer scholarships which give a higher amount of dollar value and higher number of awards (where the students have higher possibility to get the award). The priority of the scholarship from the committee is used as the tie-breaker to avoid ties in the student ranking process.

For the instance I, with a set of scholarships S and a set of students T, the ranking process ranks the students in the set of D_i for scholarship s_i (from 1 to $|D_i|$), where $r_{s_i}(t_j)$ is the rank of t_j in D_i. The ranking process also ranks the scholarship in the set of H_j for student t_j (from 1 to $|H_j|$), with $r_{t_j}(s_i)$ is the rank of s_i in H_j.

The results of the ranking process are sets of strictly ordered ranks of students for each scholarship and sets of strictly ordered ranks of scholarships for each student.

3.2.4 Optimized Matching

This process matches scholarships with students. A scholarship can be matched to several students (based on the number of the awards the scholarship can give). A student can only be awarded at most one scholarship.

For the instance I, with a set of S scholarships and a set of T students, an assignment M is a subset of $S \times P$ such that $(s_i, t_j) \in M$, implies that $t_j \in D_i$. If $(s_i, t_j) \in M$, we can say that s_i is assigned to t_j and t_j is assigned to s_i. For each scholarship $s_i \in S$, $M(s_i)$ denotes the set of students assigned to s_i in M. For each student $t_j \in T$, $M(t_j)$ denotes the set of scholarship assigned to t_j in M.

A matching M is an assignment that satisfying the following conditions:

1. For each $s_i \in S$, $|M(s_i)| \leq d_i$, and
2. For each $t_j \in T$, $|M(t_j)| \leq 1$.

To find the optimal matching, we apply two approaches. The approaches are One-Sided Matching and Two-Sided Matching.

3.3 Methods for Optimized Matching

The scholarship-student matching can be modeled as bipartite matching. It can be represented by bipartite graph G (S, T, E), where S is the vertices that represent the set of scholarships and T is the vertices that represent the set of students. I apply two approaches to optimize the matching process, a one-sided match and a two-sided match. To further explain the two approaches, we use an example instance of the Scholarship-Student Matching with a set of scholarships $S = \{s_1, s_2, s_3\}$ and a set of students $T = \{t_1, t_2, t_3, t_4, t_5\}$. Figure 9 shows the bipartite graph for the above instance.

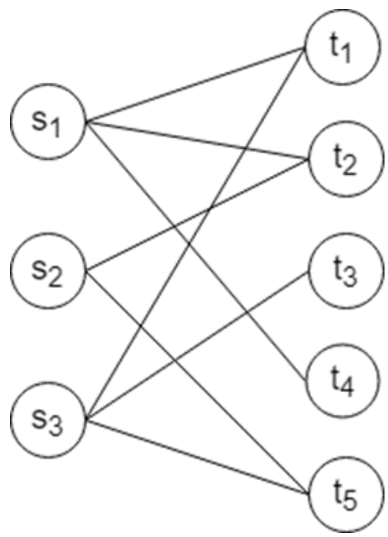

Figure 9: Bipartite graph for an instance of scholarship-student matching problem

3.3.1 One-Sided Matching

This approach considers the preferences of the scholarships over students, which represented by the ranking of students in each scholarship. The objective of this approach is to maximize the number of matches between scholarships and students with the students having the highest rank receiving the awards. This approach can be formulated as a network flow problem, where the flow of the network is the possible assignment of scholarships to students. Figure 10 shows the network flow model for the example instance.

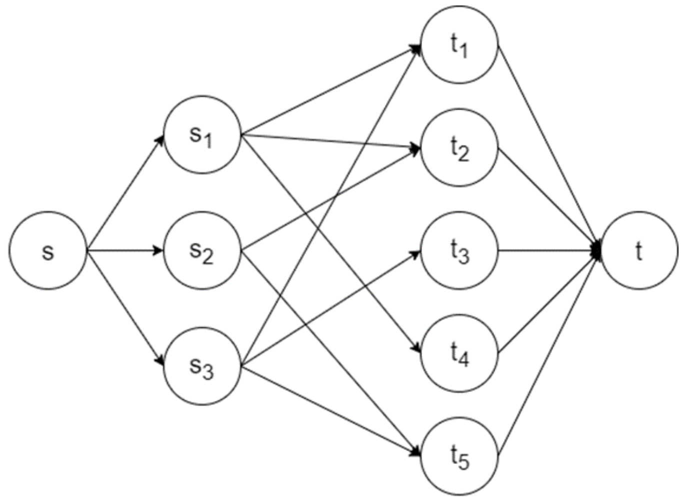

Figure 10: The network flow model for the example instance

The network is created from the bipartite graph in Figure 10 with the addition of two vertices {s, t}, where s is the source and t is the sink. Each scholarship and student is represented as a vertex. Directed edges connect source to each scholarship and each scholarship to the qualified students. Directed edges are added from each student to sink. Each edge has a capacity and a cost. All the edges have a capacity, with the lower bound value of zero, and the upper bound of one. However, the edges from source to scholarship have different upper bounds for the capacity. For the edge from source to scholarship s_i, it has the upper bound capacity value of p_i (the number of awards allocated for scholarship s_i). The cost for the edges from source to scholarship is the priority of the scholarship. For the edge from scholarship s_i to student t_j, the cost is $r_{s_i}(t_j)$, which is the rank for student t_j in scholarship s_i. While the cost for the edges from all students to sink is zero.

This problem can be considered as maximum flow minimum cost because we want to maximize the number of awards from scholarships that can be given (flow) and minimize the sum of the rankings of students who received scholarships (cost). In this research, the one-sided matching approach is solved with the help of NetworkX package in Python (Hagberg, Swart, & S Chult, 2008).

Table 2 shows the priority, the number of awards, and the list of qualified students for each scholarship for the example instance we defined above. The ranking for each student on the list of qualified students is inside the parenthesis.

Table 2: Scholarship data for the example instance

Scholarship	Priority	Number of awards	Qualified students
s1	1	2	t1 (1), t2 (2), t4 (3)
s2	2	1	t2 (1), t5 (2)
s3	3	1	t1 (1), t3 (2), t5 (3)

Figure 11 shows the network from the example instance defined above with the upper bound capacity and cost for each edge based on the data from Table 1.

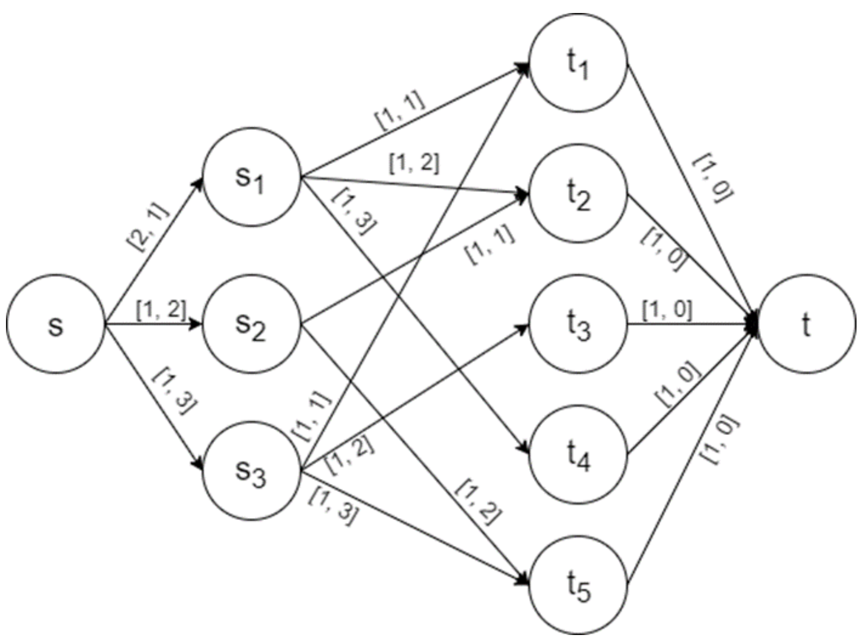

Figure 11: The network for the example instance with capacity and cost for each edge

The results for the example instance solved with maximum flow minimum cost is shown in Figure 12.

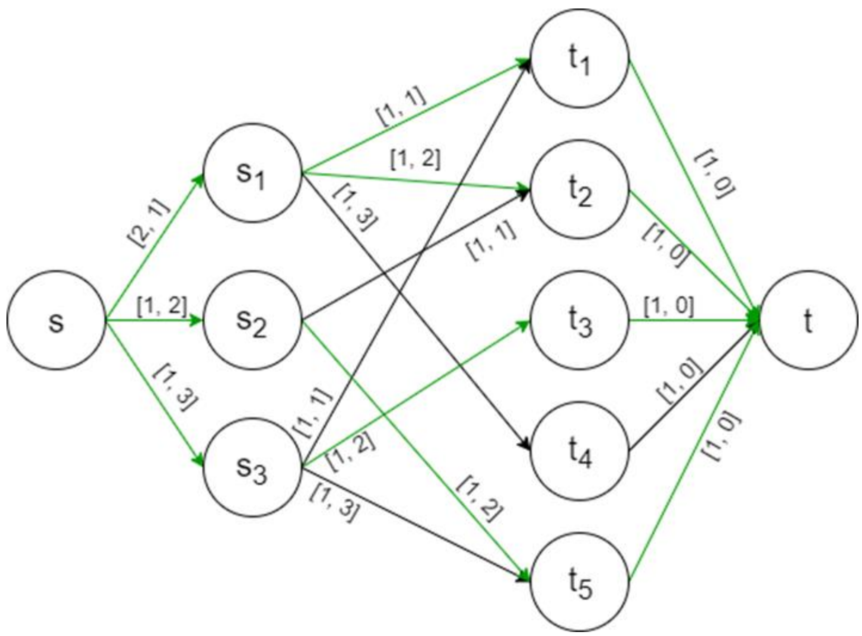

Figure 12: The maximum flow with minimum cost for the example instance

The edges in green show the flow in the network. The maximum flow for example instance is 4 (because the total allocated award from the scholarships is 4), while the minimum cost from the maximum flow is 14. The minimum cost is the sum of the result from multiplying the cost and the capacity for the green color edges. The matching result for the example instance is in Table 3.

Table 3: The matching result for the example instance for one-sided matching

Scholarship	Number of awards	Assigned students
s1	2	t1, t2
s2	1	t5
s3	1	t3

3.3.2 Two-Sided Matching

This approach considers the preferences of both sides, the scholarships and the students. The preferences of scholarships over students are represented by the ranking of students for each scholarship, while the preferences of students over scholarships are represented by the ranking of scholarship for each student. The objective of this approach is to maximize the value of the scholarship award for the student given that the student meets the scholarship requirements. This approach can be formulated as a one-to-many stable matching problem.

A matching between scholarship and student is stable when there are no blocking pairs. For the instance I, with a set of scholarships S and a set of students T, a blocking pair is an assignment of (s_i, t_j) which is not a subset of M, where:

1. t_j is in D_i, means that student t_j is qualified for scholarship s_i
2. either t_j is unmatched, or $r_{t_j}\left(M(t_j)\right)$ is lower than $r_{t_j}(s_i)$, means that student t_j is unmatched in matching M, or student t_j is matched, but the ranking of M(t_j) – which is the scholarship assigned to student t_j is lower than scholarship s_i.
3. either s_i is undersubscribed in M or the lowest of $r_{s_i}(M(s_i))$ is lower than $r_{s_i}(t_j)$, means that the awards for scholarship s_i are not fully assigned to students or the lowest ranking of students assigned to scholarship s_i in matching M is lower than student t_j.

This scholarship-student matching is a variant of a Hospital-Resident (HR) problem. Like HR, there are two algorithms based on which set's ranking is considered first, the scholarship-oriented or the student-oriented.

For the student-oriented algorithm, the students apply for the scholarship in their list based on the ranking of the scholarships for each student. The steps for matching are described as follow:

1. At the start of the algorithm, all students and scholarships are free.
2. The students are assigned to the scholarship based on the rank of the scholarships matched to the students.
3. If there is an unallocated award in a scholarship, the scholarship is awarded to the student with the highest rank. If the award has been assigned, but there is another student who has higher rank on the scholarship list, the award is re-allocated to a higher rank student. The previous student is un-assigned to the scholarship.
4. Step (2) and (3) are repeated until all scholarship awards are allocated or there are no students remaining which meet the requirements for the scholarship.

For the scholarship-optimal match, the scholarships assign the students in their list based on the ranking of the students in each scholarship. The steps for matching describe as follow:

1. At the start of the algorithm, all students and scholarships are free.
2. The scholarships are assigned to the students based on the rank of the students matched to the scholarship.
3. If the students have not been allocated a scholarship, the student accepts the scholarship with the highest rank on the student's list. If the students are assigned to another scholarship that has a higher rank on the

student's list, the student accepts the new scholarship. The previous scholarship is un-assigned to the student.

4. Step (2) and (3) are repeated until all scholarship awards are allocated or there are no students remaining which meet the requirements for the scholarship.

For the example instance defined at the beginning of the section 3.3, Table 4 shows the number of awards and the ranking of students for each scholarship, and Table 5 shows the ranking of scholarships for each student. The ranking is inside the parenthesis. The result of the matching with the two-sided approach is shown in Table 6.

Table 4: The number of awards and the ranking of students for each scholarship

Scholarship	Number of awards	Qualified students
s1	2	t1 (1), t2 (2), t4 (3)
s2	1	t2 (1), t5 (2)
s3	1	t1 (1), t3 (2), t5 (3)

Table 5: The ranking of scholarships for each student

Student	Scholarship
t1	s1 (1), s3 (2)
t2	s1 (1), s2 (2)
t3	s3 (1)
t4	s1 (1)
t5	s2 (1), s3 (2)

Table 6: The matching result for the example instance with two-sided approach

Scholarship	Student	
	Student-oriented	Scholarship-oriented
s1	t1, t2	t1, t2
s2	t5	t5
s3	t3	t3

For the example instance, the matching result for the student-oriented and the scholarship-oriented are identical. This can happen, but this will not, in general, be the case (Gusfield & Irving, 1989).

Chapter 4: Result

The approaches described in the previous chapter are tested with a dataset consisting of 95 scholarship funds with 333 total allocated awards and 923 student applicants. The detailed description of the dataset is in chapter 3 section 3.1.3. This chapter describes the results of the testing. The structure of this chapter is as follows; section 4.1 describes the metrics which are used to evaluate the matching results. In section 4.2, we describe the results of the two approaches for the student data subsets used for testing. Section 4.3 contains the comparison of the two approaches.

4.1 Metrics for Matchings

We use several metrics to explain and compare the matching results. Those metrics are the size of the match (Diebold & Bichler, 2017) and the percentage of dollar value spent from the scholarship accounts.

The first metric is the size of the match. The size of the match represents the percentage of scholarships matched to students at the end of the matching process. We use two metrics for the size of match. The first one, overall match, calculates the metric using the total number of awards that can be allocated. The second metric, qualified matches, calculates the metric using the minimum number of qualified students (the sum of the minimum value between the number of awards and the number of qualified students for each scholarship).

The formula for the overall match metric is as follow.

$$overall\ match = \frac{|M|}{\sum_{i=1}^{n} u_i} \times 100\%$$

where $|M|$ is the number of scholarships and students matched, and u_i is the number of allocated awards for each scholarship s_i.

The formula for qualified matches is as follow.

$$qualified\ match = \frac{|M|}{\sum_{i=1}^{n} \min(|D_i|, u_i)} \times 100\%$$

where $|M|$ is the number of scholarship-student matchings, $|D_i|$ is the number of students who qualify for scholarship s_i, and u_i is the allocated number of award for each scholarship s_i.

The difference between these two affects the interpretation of the size of the matching. We see the difference between them in the result explanation.

The second metric represents the payout from the scholarship fund. This is measured as the percentage of available scholarship funds spent. This metric is calculated using two baselines. The first metric, overall payout, is based on the total dollar value in the account, whereas the second metric, actual payout, is based on the total value of funds that can be spent.

Payout as measured using the first baseline is as follow.

$$overall\ payout = \frac{\sum_{i=1}^{n}(a_i \times |M_i|)}{\sum_{i=1}^{n} A_i} \times 100\%$$

where a_i is the amount of award for scholarship s_i, $|M_i|$ is the number of students matched to scholarship s_i, and A_i is the total dollar value in the account of scholarship s_i.

Payout as measured using the second baseline is as follow.

$$qualified\ payout = \frac{\sum_{i=1}^{n}(a_i \times |M_i|)}{\sum_{i=1}^{n} \lfloor A_i \rfloor} \times 100\%$$

where a_i is the amount of award for scholarship s_i, $|M_i|$ is the number of students matched to scholarship s_i, and $\lfloor A_i \rfloor$ is the total amount of dollar value in the account of scholarship s_i rounded down into the nearest thousand values.

4.2 Matching Results

I describe the results for both approaches in this section. The result for one-sided matching is in section 4.2.1, and the result for the two-sided matching is in section 4.2.2.

4.2.1 Result of One-Sided Matching

Table 7 shows the metrics for the one-sided matching result.

Table 7: Metrics for one-sided matching for three datasets

Dataset	Overall Match (%)	Qualified Match (%)	Overall Payout (%)	Actual Payout (%)
full	95.19	100	87.85	94.49
sample 1	85.28	100	72.64	78.14
sample 2	90.39	100	80.17	86.23

For one-sided matching, the overall match value for the full dataset is 95.19%, for sample 1 is 85.28%, and for sample 2 is 90.39%. However, the qualified match value for all datasets is 100%. The difference on these two metrics is because for a scholarship s_i, with d_i allocated award, the number of students who qualified for s_i is less than d_i. This condition decreases the total number of awards that can be allocated to students. The qualified match value shows that the result fulfilled the objective of the approach, which is to maximize the number of allocated award received by the students.

The different values for the overall match between the full dataset and the sample (1 and 2) dataset is because of the different number of matches produced for each dataset. The full dataset has a higher number of matches than both sample datasets. The reason is because there are more qualified students in the full dataset than the sample datasets, while the number of allocated awards are the same for all datasets. The values of qualified match are the same for all dataset.

The overall payout for the full dataset is 87.85%, for sample 1 is 72.64%, and for sample 2 is 80.17%. This means that the total value of the awards given to students was less than the dollar value of the account. This is partly because of the process of rounding down the dollar values in the accounts when the committee calculates the amount of funds to be spent for each scholarship.

The actual payout for the full dataset is 94.49%, for sample 1 is 78.14%, and for sample 2 is 86.23%. This again indicates that the total amount of awards was less than the funds allocated for the scholarships. This happens because there are scholarships where the number of qualified students is less than the number of the allocated awards. This condition affects the total amount of award spent. This condition indirectly affects the overall payout, because it decreases the total amount of award given to students.

The ranking process ranks the students based on their GFN and GPA. Figure 13, Figure 14 and Figure 15 show the histogram of one-sided matching results for the number of students receiving awards by their GFN for the three datasets. The histograms of the three datasets are skewed to higher gross financial need, as would be expected.

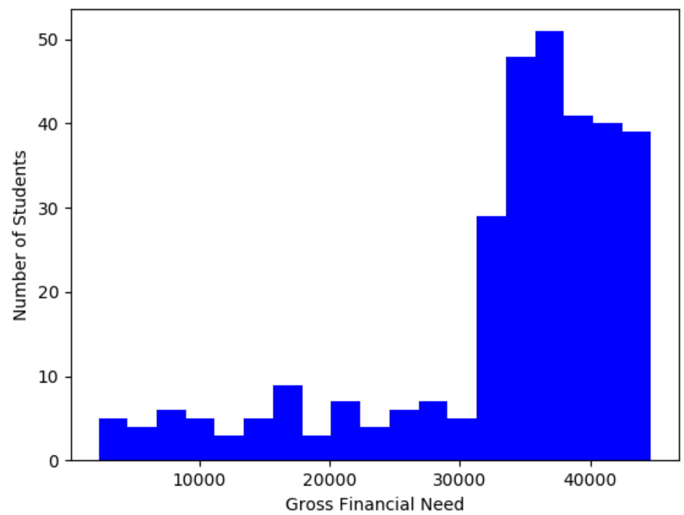

Figure 13: Histogram of GFN for one-sided matching result for full dataset

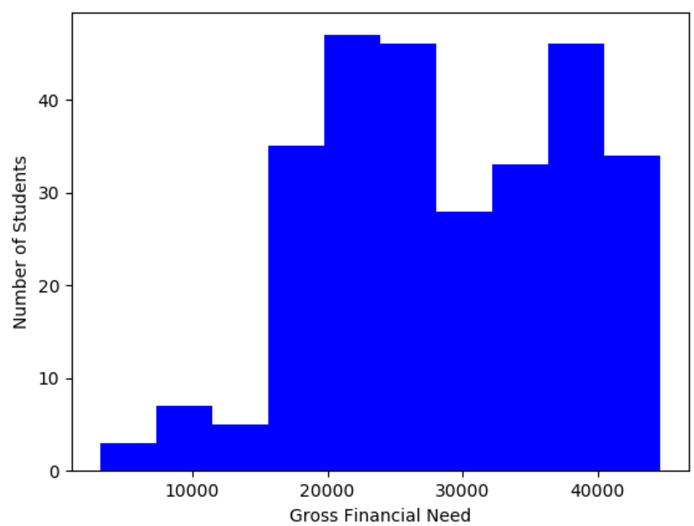

Figure 14: Histogram of GFN for one-sided matching result for sample 1 dataset

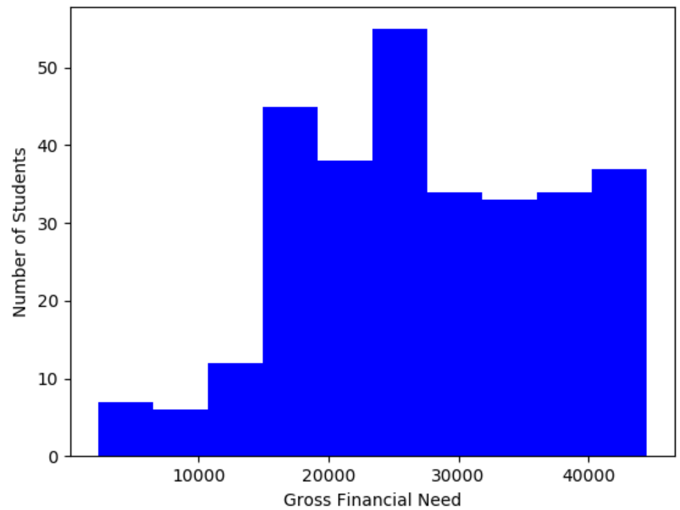

Figure 15: Histogram of GFN for one-sided matching result for sample 2 dataset

Figure 16, Figure 17 and Figure 18 show the histogram of one-sided matching results for the number of students receiving awards by their GPA for the three datasets. The histograms of the three datasets are skewed to the right. The effect of filtering the data subsets by GPA is clearly shown in Figures 17 and 18 and it is clear that the highest GPAs are receiving more scholarships.

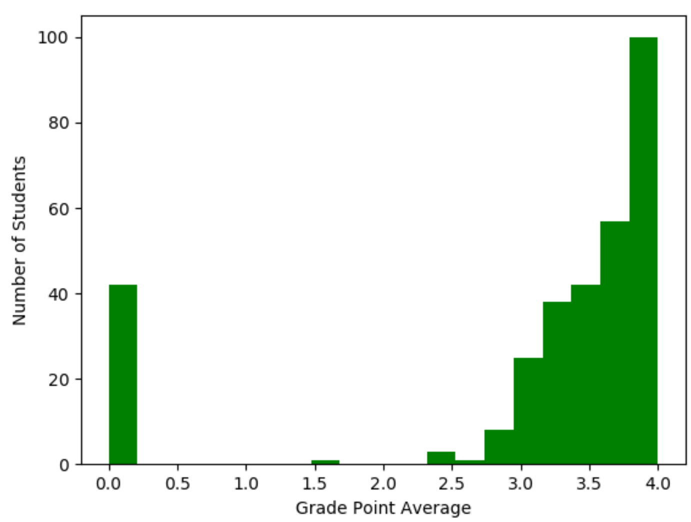

Figure 16: Histogram of GPA for one-sided matching result for full dataset

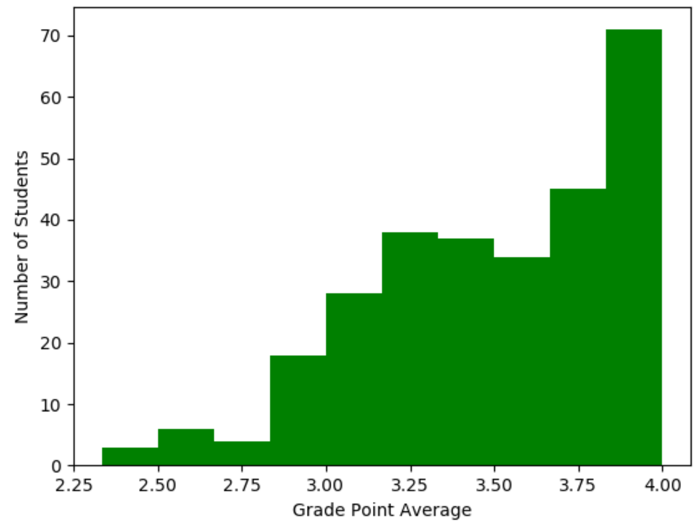

Figure 17: Histogram of GPA for one-sided matching result for sample 1 dataset

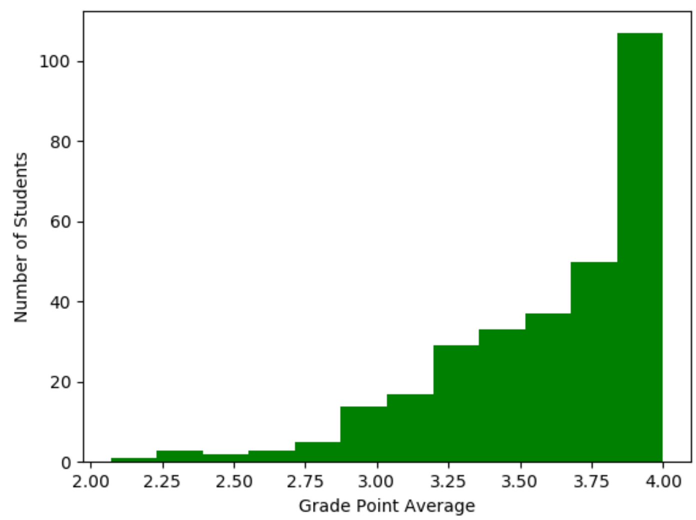

Figure 18: Histogram of GPA for one-sided matching result for sample 2 dataset

Table 8 provides the summary statistics of GFN for three datasets.

Table 8: Summary statistics of GFN for one-sided matching result for three datasets

Dataset	Min	Median	Mean	Max
full	2,250	36,608	33,579	44,698
sample 1	3,144	27,944	28,877	44,663
sample 2	2,250	26,942	27,026.45	44,517

The lowest GFN of the student who receives a scholarship is $2,250 (for full and sample 2 datasets) and $3,144 for sample 1 dataset. This can happen when the student qualified for a scholarship and there is no other student who qualified for the scholarship with higher GFN. The values of maximum GFNs of the student who receive a scholarship is the same as the highest value of GFN for each of the dataset. This happens because we prioritize student with higher GFN to get the scholarship in the ranking process.

Table 9 provides the summary statistics of GPA for three datasets.

Table 9: Summary statistics of GPA for one-sided matching result for three datasets

Dataset	Min	Median	Mean	Max
full	0	3.54	3.07	4
sample 1	2.33	3.54	3.5	4
sample 2	2.07	3.69	3.59	4

The lowest GPA of students who can receive a scholarship for the full dataset is 0. The value of 0 for GPA caused by the missing data values. However, there are students with a GPA of 0 who can still get an award because there are several scholarships which do not include GPA as a criterion. Given that this is not the desired result for the matching, we can remove the students with a GPA of 0 filtering out GPA below 2.0. As shown, the minimum GPA values for sample 1 and sample 2 dataset are 2.33 and 2.07, respectively.

4.2.2 Result of Two-Sided Matching

Table 10 shows the metrics for the two-sided matching result.

Table 10: Metrics for two-sided matching results for three datasets

Dataset	Approach	Overall Match (%)	Qualified Match (%)	Overall Payout (%)	Actual Payout (%)
full	Scholarship-oriented	94.89	99.68	87.37	93.97
	Student-oriented	94.89	99.68	87.37	93.97
sample 1	Scholarship-oriented	84.38	98.94	71.52	76.93
	Student-oriented	84.38	98.94	71.52	76.93
sample 2	Scholarship-oriented	89.48	99	79.04	85.02
	Student-oriented	89.48	99	79.04	85.02

For the two-sided approach, both the scholarship-oriented and student-oriented methods, the overall match values for each dataset are the same. The overall match for the full dataset is 94.89%. However, the qualified match value is 99.68%. The difference of these two metrics is because for a scholarship s_i, with d_i allocated award, the number of students who qualified for s_i is less than d_i. This condition decreases the total number of awards that can be allocated to students.

The overall payout, for both scholarship-oriented and student-oriented, is 87.37% for the full dataset, 71.52% for sample 1, and 79.04% for sample 2. The value means that the total amount of awards given to students is less than the dollar value of the account. This is partly because of the process of rounding down the dollar values in the accounts when the committee calculates the amount of funds to be spent for each scholarship. The overall payout for scholarship-oriented and student oriented are the same because they both have the same number of scholarship-student match.

Both the scholarship-oriented and student-oriented methods for each dataset yield the same actual payout value. It happens because they both have the same number of scholarship-student matches. The actual payout value is 93.37% for the full dataset, 76.93% for sample 1, and 85.02% for sample 2. This indicates that the total amount of awards is less than the funds allocated for the scholarships. This happens because there are scholarships where the number of qualified students is less than the number of the allocated awards. This condition affects the total amount of award spent. This condition indirectly affects the overall payout, because it decreases the total amount of award given to students.

Figure 19 and Figure 20 show the histogram of GFN for the scholarship-oriented and student oriented for the full dataset. The patterns are similar as with the one-sided match; students with higher GFN receive more awards regardless of the method.

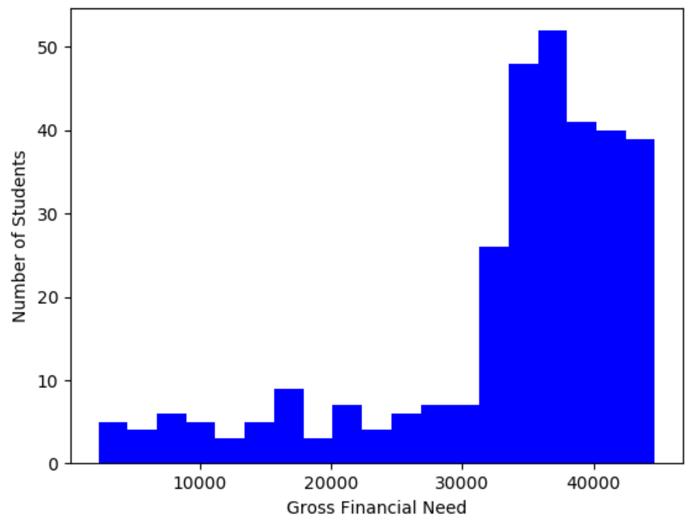

Figure 19: Histogram of GFN for two-sided scholarship-oriented for full dataset

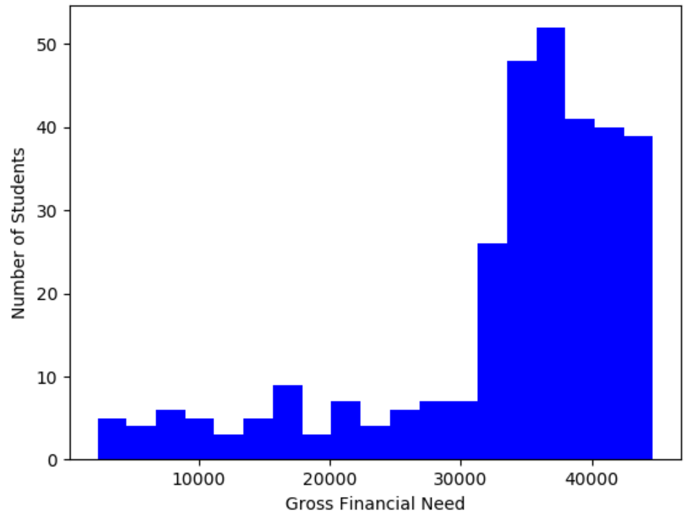

Figure 20: Histogram of GFN for two-sided student-oriented for full dataset

Figure 21 and Figure 22 show the histogram of GFN for the scholarship-oriented and student oriented for sample 1 dataset. Figure 23 and Figure 24 show the histogram of GFN for the scholarship-oriented and student oriented for sample 2 dataset. The histograms of GFN are skewed to the right. The histograms of GFN for scholarship-oriented and student-oriented for each dataset are identical.

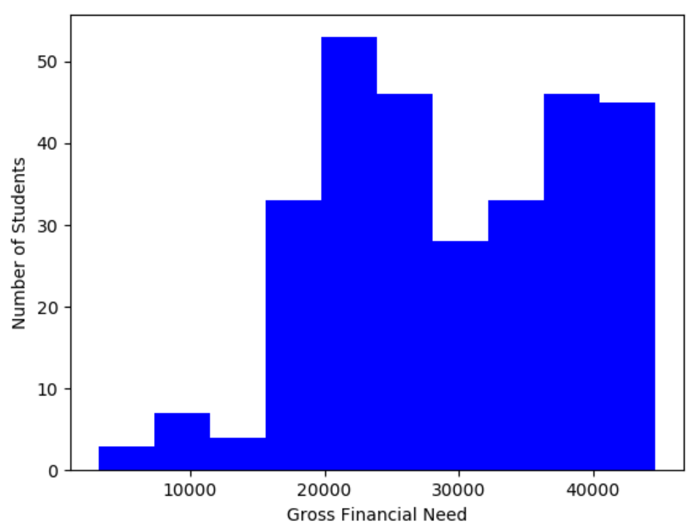

Figure 21: Histogram of GFN for two-sided scholarship-oriented matching for sample 1 dataset

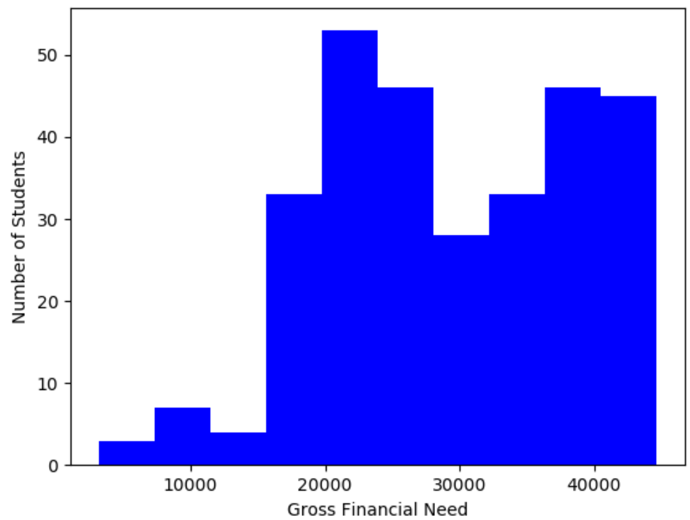

Figure 22: Histogram of GFN for two-sided student-oriented matching for sample 1 dataset

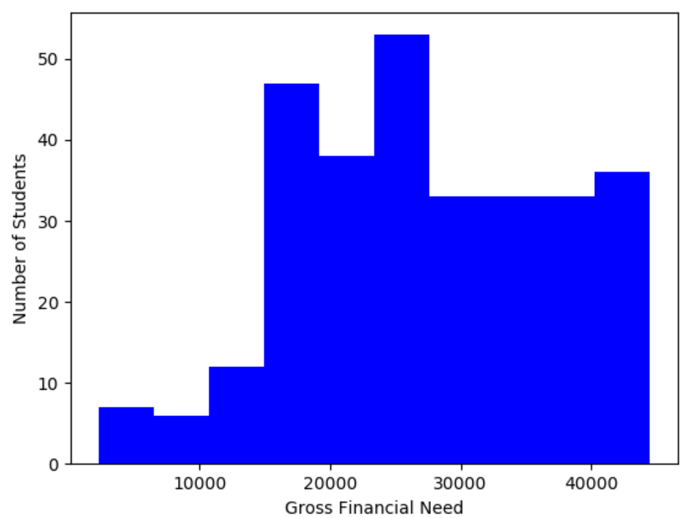

Figure 23: Histogram of GFN for two-sided scholarship-oriented matching for sample 2 dataset

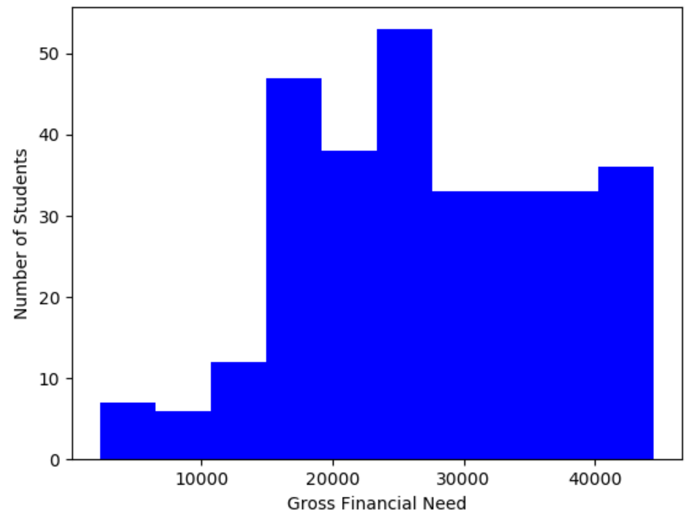

Figure 24: Histogram of GFN for two-sided student-oriented matching for sample 2 dataset

Figure 25 and Figure 26 show the histogram of GPA for the scholarship-oriented and student oriented for the full dataset.

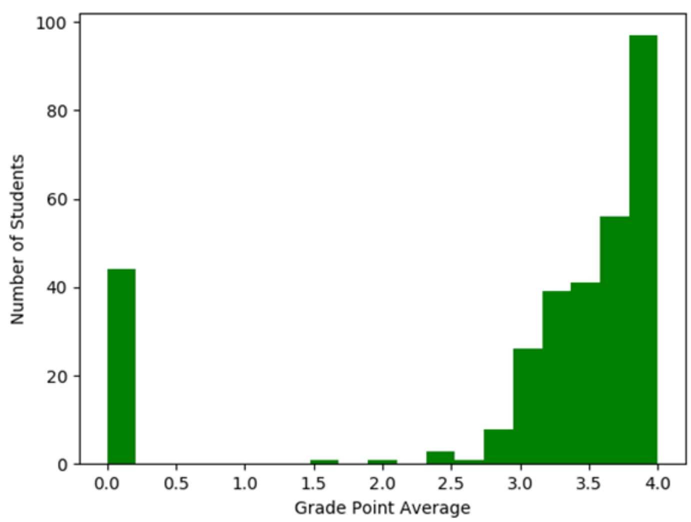

Figure 25: Histogram of GPA for two-sided scholarship-oriented matching for full dataset

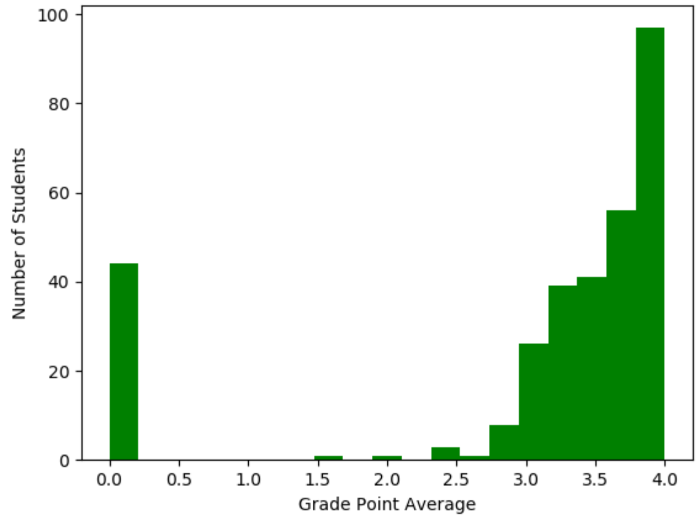

Figure 26: Histogram of GPA for two-sided student-oriented matching for full dataset

Figure 27 and Figure 28 show the histogram of GPA for the scholarship-oriented and student oriented for sample 1 dataset.

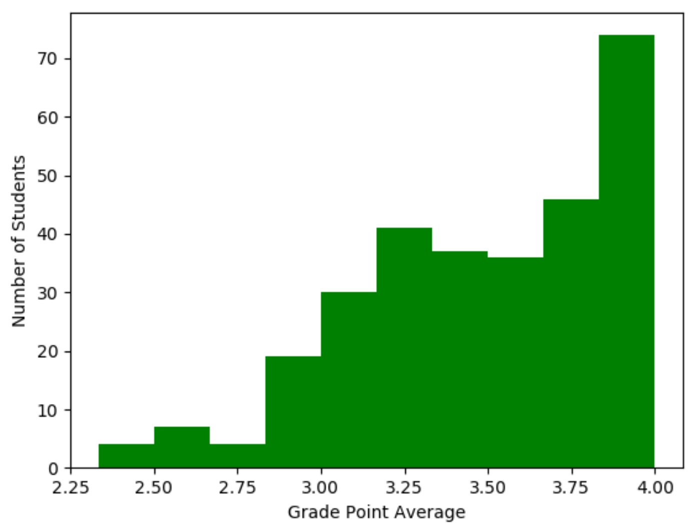

Figure 27: Histogram of GPA for two-sided scholarship-oriented matching for sample 1 dataset

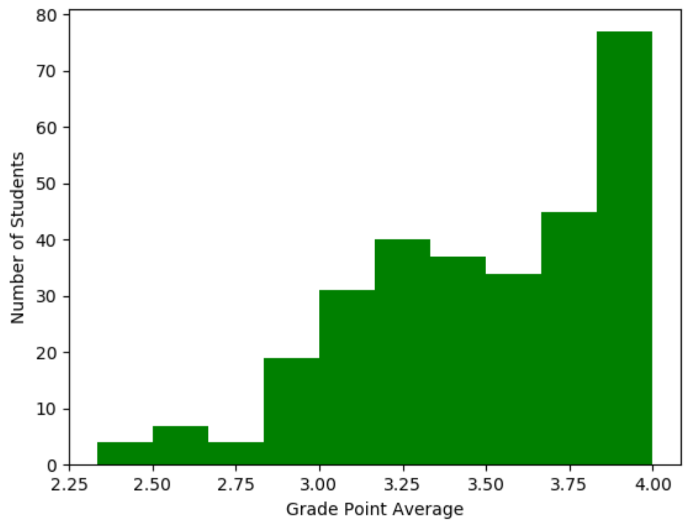

Figure 28: Histogram of GPA for two-sided student-oriented matching for sample 1 dataset

Figure 29 and Figure 30 show the histogram of GPA for the scholarship-oriented and student oriented for sample 2 dataset.

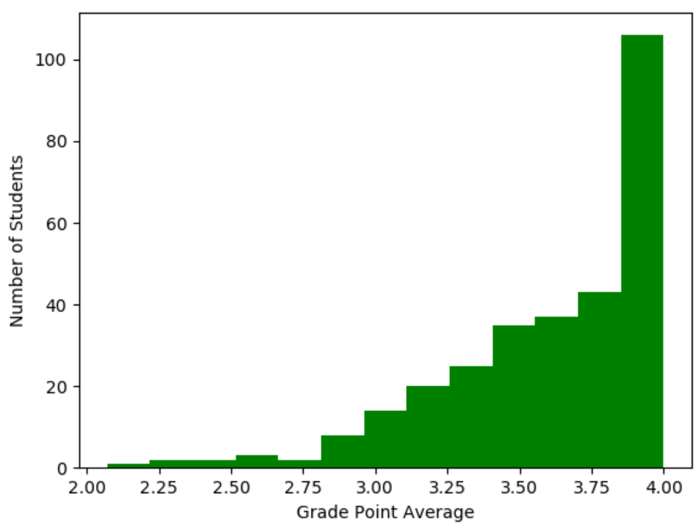

Figure 29: Histogram of GPA for two-sided scholarship-oriented matching for sample 2 dataset

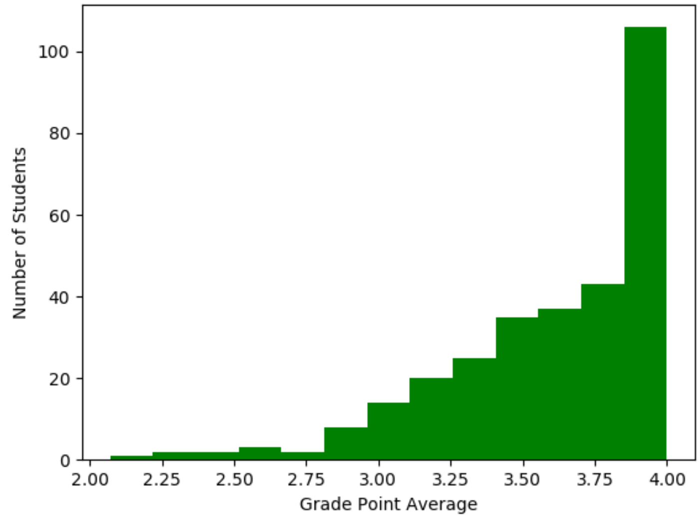

Figure 30: Histogram of GPA for two-sided student-oriented matching for sample 2 dataset

The histograms of GPA are skewed to the right. The histogram of GPA for the scholarship-oriented and student-oriented for each dataset are identical. The histogram skewed to the right means that the students who received the scholarship are from the higher range GPA.

Table 11 shows the summary statistics for GFN of the two-sided matching result for the datasets. The value for each statistic (Min, Median, Mean, Max) for GFN is the same between the scholarship-oriented and student-oriented for each dataset.

Table 11: Summary statistics for GFN of the two-sided matching results

Dataset	Approach	Min	Median	Mean	Max
full	Scholarship-oriented	2,250	36,636	33,590	44,698
	Student-oriented	2,250	36,636	33,590	44,698
sample 1	Scholarship-oriented	3,144	28,209	29,003	44,663
	Student-oriented	3,144	28,209	29,003	44,663
sample 2	Scholarship-oriented	2,250	27,050	27,156	44,517
	Student-oriented	2,250	27,051	27,156	44,517

Table 12 shows the summary statistics for GPA of the two-sided matching results. The value for each statistic (Min, Median, Mean, Max) for GPA is the same between the scholarship-oriented and student-oriented for each dataset.

Table 12: Summary statistics for GPA of the two-sided matching result

Dataset	Approach	Min	Median	Mean	Max
full	Scholarship-oriented	0	3.54	3.07	4
	Student-oriented	0	3.54	3.07	4
sample 1	Scholarship-oriented	2.33	3.54	3.5	4
	Student-oriented	2.33	3.54	3.5	4
sample 2	Scholarship-oriented	2.07	3.7	3.6	4
	Student-oriented	2.07	3.7	3.6	4

Based on the metrics, the histograms for GFN and GPA, and the statistics for GFN and GPA, we can say that the matching results for scholarship-oriented and student-oriented are similar assigned the scholarships to the same subset of students.

4.3 Comparison

Table 13 shows the summary of the metrics for all approaches.

Table 13: Summary of metrics for all approaches

Dataset	Metric	One-sided	Two-sided Scholarship-oriented	Two-sided Student-oriented
full	Overall Match (%)	95.19	94.89	94.89
	Qualified Match (%)	100	99.68	99.68
	Overall Payout (%)	87.85	87.37	87.37
	Actual Payout (%)	94.49	93.97	93.97
sample 1	Overall Match (%)	85.28	84.38	84.38
	Qualified Match (%)	100	98.94	98.94
	Overall Payout (%)	72.64	71.52	71.52
	Actual Payout (%)	78.14	76.93	76.93
sample 2	Overall Match (%)	90.39	89.48	89.48
	Qualified Match (%)	100	99	99
	Overall Payout (%)	80.17	79.04	79.04
	Actual Payout (%)	86.23	85.02	85.02

For both metrics, overall match and qualified match, the one-sided matching gives a higher number of matches than the other approaches. although, the difference is negligible. The difference is because the objective of the one-sided approach is to maximize the number of scholarships assigned to students. The same situation also happens for the metrics. overall payout and qualified payout. (. The one-sided matching gives a higher percentage. This is directly related to the match metrics, because the one-sided approach produces more matching than the other approaches.

Table 14 shows the summary statistics of GFN for all the approaches.

Table 14: Summary statistics of GFN for all approaches

Dataset	Statistic	One-sided	Two-sided Scholarship-oriented	Two-sided Student-oriented
full	Min	2,250	2,250	2,250
	Median	36,608	36,636	36,636
	Mean	33,579	33,590	33,590
	Max	44,698	44,698	44,698
sample 1	Min	3,144	3,144	3,144
	Median	27,944	28,209	28,209
	Mean	28,877	29,003	29,003
	Max	44,663	44,663	44,663
sample 2	Min	2,250	2,250	2,250
	Median	26,942	27,051	27,051
	Mean	27,026	27,156	27,156
	Max	44,517	44,517	44,517

The differences between the one-sided and two-sided approaches are only for the median and mean values. The differences are very small and are a function of the random selection of the two subsets.

Table 15 shows the summary statistics of GPA for all the approaches. The statistics of the GPA for all approaches have similar value.

Table 15: Summary statistics of GPA for all approaches

Dataset	Statistic	One-sided	Two-sided Scholarship-oriented	Two-sided Student-oriented
full	Min	0	0	0
	Median	3.54	3.54	3.54
	Mean	3.07	3.07	3.07
	Max	4	4	4
sample 1	Min	2.33	2.33	2.33
	Median	3.54	3.54	3.54
	Mean	3.5	3.5	3.5
	Max	4	4	4
sample 2	Min	2.07	2.07	0.12
	Median	3.70	3.71	3.71
	Mean	3.60	3.61	3.61
	Max	4	4	4

Appendix C, D, and E show the total and average amount of awards based on the category of GFN and GPA for the full and sample datasets, respectively. Appendix F and G show the number of awards for n-rank students, and the number of students who received n-rank scholarship, both with the full dataset, respectively. From the appendices, we can see that the number of awards given to students with certain rank are varied. The variations are hard to be seen in the higher rank. It happens after the 30th rank. These variations make the assignment of scholarships to students different for each approach.

Based on the two metrics and the statistics of GFN and GPA, we can say that there is not much of a difference between the approaches. Although, based on the metrics, one-sided approach gives a slightly better result. However, it should be noted that the slight difference in here is equal to a student who does not get assigned a scholarship in the two-sided approach.

4.4 Application

I developed an application for the two approaches to facilitate use by the scholarship committee. The application is developed in Python with a web interface. Figures 31, 32, and 33 show the example of the web interface for the initial matching, one-sided matching, and two-sided matching respectively. The committee can use this application to perform the matching process. The result of the matching process is displayed on the interface. The result then can be saved in an Excel file. An example of the result from one-sided matching process which is saved in an Excel file can be seen in Appendix H. The committee can use the matching result to assign the scholarships to the students.

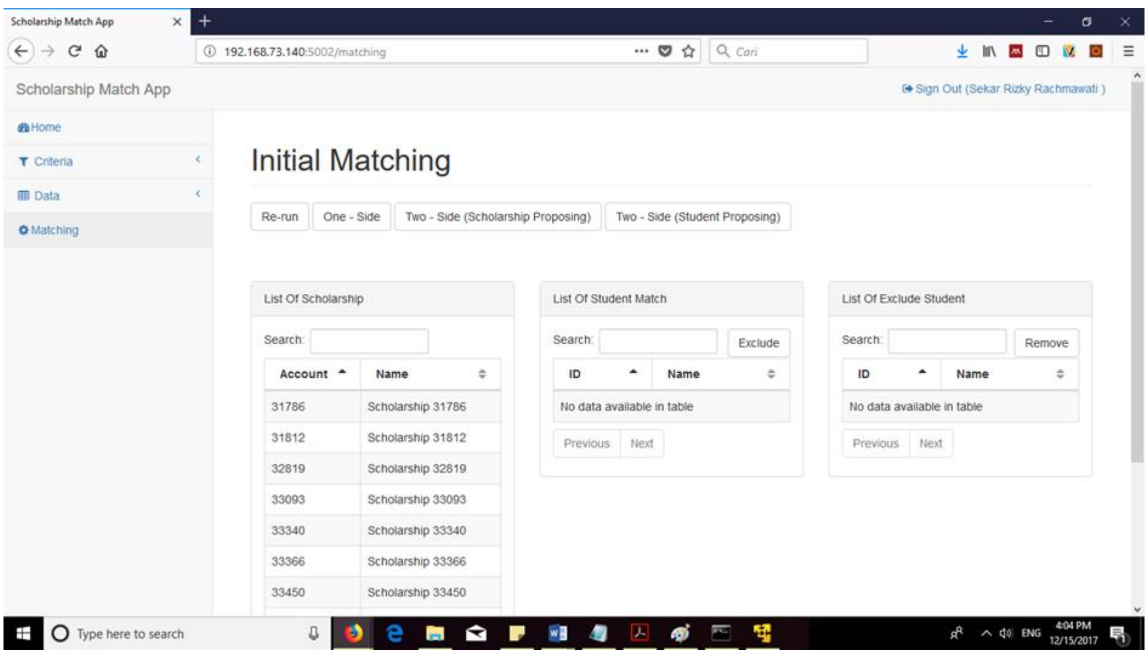

Figure 31:Interface for the initial matching

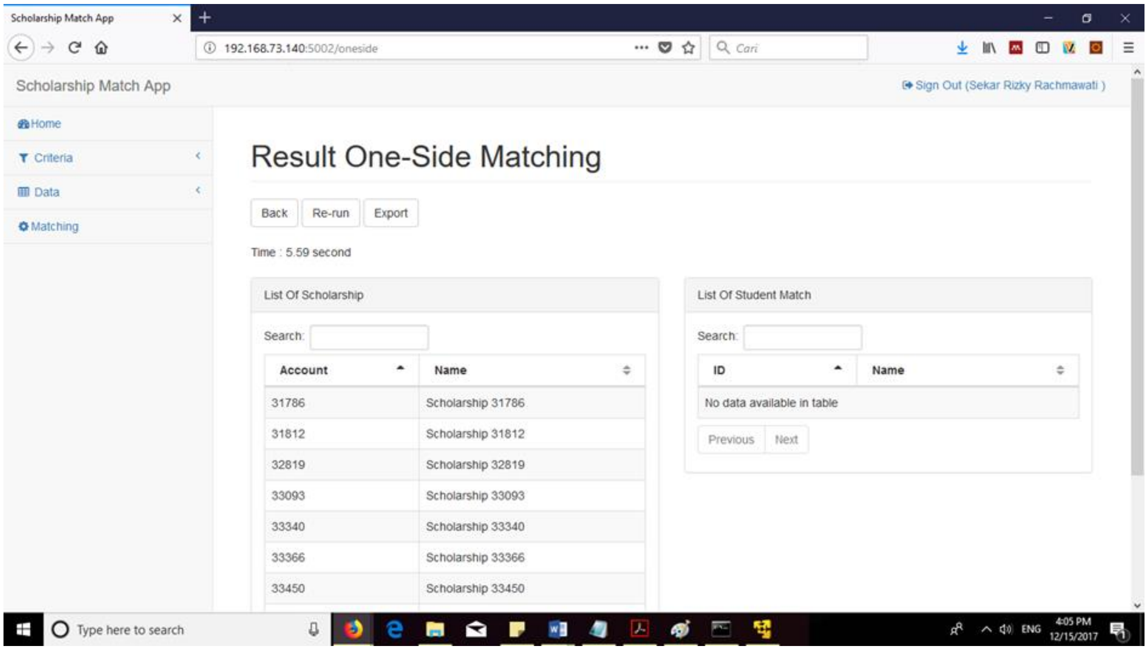

Figure 32: Interface for one-sided matching

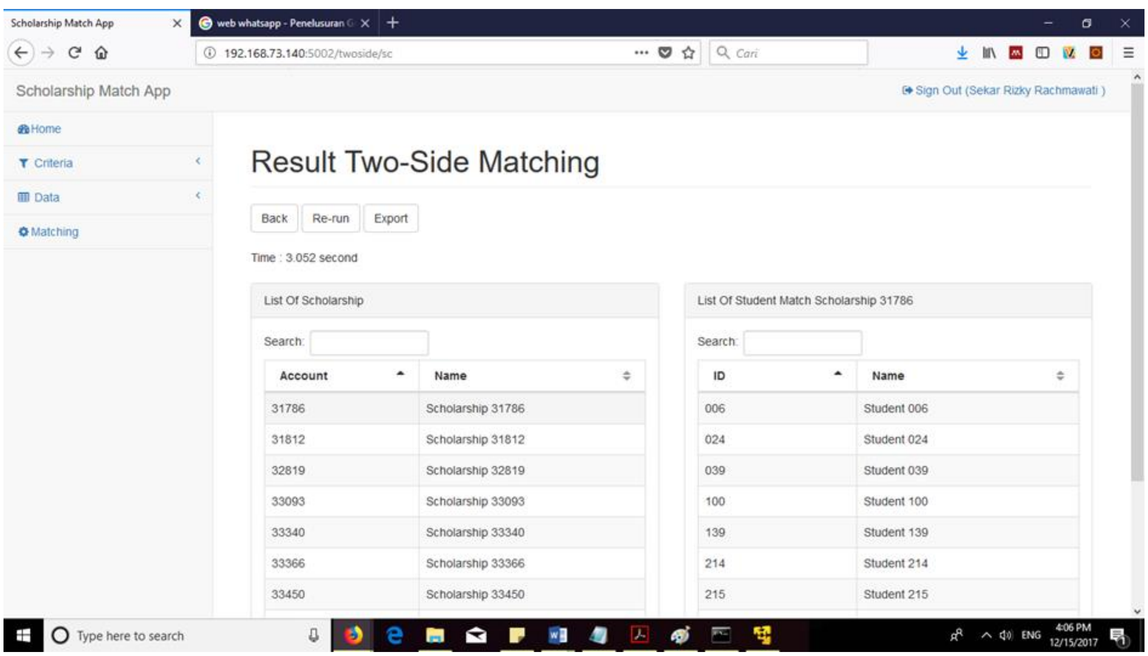

Figure 33: Interface for two-sided matching

Chapter 5: Conclusion

In this thesis, I propose two approaches to optimize the scholarship-student matching process for GCoE. Those approaches are one-sided matching, based on the ranking of students who met the requirements for each scholarship, and two-sided matching, which maximizes the value of award the students can get. The approaches are applied to an actual dataset of scholarships and students.

The results show that the approaches produce viable scholarship-student matching sets. The matching assigned the scholarships to students. The one-sided approach is slightly better than the two-sided approach because it assigns all the allocated award to eligible students.

There are several areas in which this work can be extended, which include assigning a different award amount for a scholarship, applying different ranking methods, and including several attributes of the scholarships which are excluded in this work. These recommendations are based on the assumptions and limitations of this work.

This work uses the assumption that the amount of award given to students for a particular scholarship is the same between all the students who are assigned to that particular scholarship. It can be improved by allowing different students to receive a different award amounts based on criteria specified by the committee member. This improvement will help the committee to tailor the amount of award based on the individual financial need of the students, which will make the students receive the amount of award closer to the value of their financial need.

The current ranking process for students is based on two attributes, Gross Financial Need (GFN) and Grade Point Average (GPA). This ranking concept needs to be re-examined, whether this is the best combination of attributes to rank the students or if there can be other combinations which will rank the students in a better way.

The current ranking process for scholarships is based on the priority set by the committee, the amount of award, and the number of awards, and a tie-breaker. Similarly with the ranking concept for the students, the ranking process for the scholarships needs to be re-examined to determine whether this ranking process is the best way. It needs to be evaluated to see if there can be another way to rank the scholarships or if it would be better not to rank the scholarships.

The current ranking process for both the scholarship and the students is strictly ordered. It can be improved by allowing ties in the ranking process so that the students with the similar qualification have the same chance to be assigned to a scholarship, not dependent on a random tie breaking.

This work excludes several attributes from the scholarship matching process. It can be improved by including those excluded attributes. To include those attributes, several methods have to be developed to extract the value for those attributes from student data. For example, there is an attribute for organization activity. To determine student participation in an organization, we need to extract that information from the student's essay. It can be achieved by developing a method using text analytics.

In conclusion, scholarship-student matching can be optimized with these two approaches. This optimized process helps the committee of the scholarships in term of reducing the time to work on the matching process. I would recommend the one-sided approach to optimize the matching process because it can allocate the possible maximum number of awards to the students.

References

Abdulkadiroğlu, A., Pathak, P. A., & Roth, A. E. (2005). The new york city high school match. *American Economic Review*, 364-367.

Abdulkadiroğlu, A., Pathak, P. A., Roth, A. E., & Sönmez, T. (2005). The Boston public school match. *American Economic Review*, 368-371.

Abdulkadiroglu, A., & Sönmez, T. (2003). School choice: A mechanism design approach. *The American Economic Review, 93*(3), 729-747.

Abraham, D. J. (2003). *Algorithmics of two-sided matching problems.*

Abraham, D. J., Irving, R. W., & Manlove, D. F. (2003). The student-project allocation problem. *Lecture notes in computer science*, 474-484.

Ahuja, R. K., Magnanti, T. L., & Orlin, J. B. (1993). Network flows: theory, algorithms, and applications.

Aksoy, S., Azzam, A. A., Coppersmith, C., Glass, J., Karaali, G., Zhao, X., & Zhu, X. (2013). School Choice as a One-Sided Matching Problem: Cardinal Utilities and Optimization. *arXiv preprint arXiv:1304.7413*.

Diebold, F., Aziz, H., Bichler, M., Matthes, F., & Schneider, A. (2014). Course allocation via stable matching. *Business & Information Systems Engineering, 6*(2), 97-110.

Diebold, F., & Bichler, M. (2017). Matching with indifferences: A comparison of algorithms in the context of course allocation. *European Journal of Operational Research, 260*(1), 268-282.

Diestel, R. (2005). Graph theory. 2005. *Grad. Texts in Math, 101*.

Even, S. (2011). *Graph algorithms*: Cambridge University Press.

Gale, D., & Shapley, L. S. (1962). College admissions and the stability of marriage. *The American Mathematical Monthly, 69*(1), 9-15.

Gale, D., & Sotomayor, M. (1985). Some remarks on the stable matching problem. *Discrete Applied Mathematics, 11*(3), 223-232.

Gusfield, D., & Irving, R. W. (1989). *The stable marriage problem: structure and algorithms*.

Hagberg, A., Swart, P., & S Chult, D. (2008). *Exploring network structure, dynamics, and function using NetworkX*.

Irving, R. W. (1985). An efficient algorithm for the “stable roommates” problem. *Journal of Algorithms, 6*(4), 577-595.

Irving, R. W. (1994). Stable marriage and indifference. *Discrete Applied Mathematics, 48*(3), 261-272.

Irving, R. W. (1998). *Matching medical students to pairs of hospitals: a new variation on a well-known theme.* Paper presented at the ESA.

Iwama, K., & Miyazaki, S. (2008). *A survey of the stable marriage problem and its variants.* Paper presented at the Informatics Education and Research for Knowledge-Circulating Society, 2008. ICKS 2008. International Conference on.

Manlove, D. F., Irving, R. W., Iwama, K., Miyazaki, S., & Morita, Y. (2002). Hard variants of stable marriage. *Theoretical Computer Science, 276*(1-2), 261-279.

Manlove, D. F., & O'Malley, G. (2008). Student-project allocation with preferences over projects. *Journal of Discrete Algorithms, 6*(4), 553-560.

McVitie, D. G., & Wilson, L. B. (1971). The stable marriage problem. *Communications of the ACM, 14*(7), 486-490.

Roth, A. E., & Peranson, E. (1999). *The redesign of the matching market for American physicians: Some engineering aspects of economic design*.

Roth, A. E., Sönmez, T., & Ünver, M. U. (2004). Kidney exchange. *The Quarterly Journal of Economics, 119*(2), 457-488.

Sng, C. T. S. (2008). *Efficient algorithms for bipartite matching problems with preferences.* University of Glasgow.

Snyder, T. D., de Brey, C., & Dillow, S. A. (2016). Digest of Education Statistics 2014, NCES 2016-006. *National Center for Education Statistics*.

Tanimoto, S. L., Itai, A., & Rodeh, M. (1978). Some matching problems for bipartite graphs. *Journal of the ACM (JACM), 25*(4), 517-525.

Appendix A: List of scholarship and student attributes

The list of attributes for this research are on the following table.

Number	Attributes
1	Classification
2	Hours
3	Enrollment
4	Minimum GPA
5	Minority
6	Gender
7	Major
8	US Resident
9	State Resident
10	County Resident
11	City Resident
12	Financial Need
13	Citizenship
14	High School Name
15	High School State
16	High School City
17	High School County

Appendix B: List of excluded attributes

The list of excluded attributes for this research are on the following table.

Number	Attributes
1	High SAT/ACT score
2	Active in campus organization
3	Active in extracurricular activities
4	Married
5	Total family income of less than 50K/year
6	Demonstrated leadership
7	Rural Oklahoma
8	Hometown Population 25K or less
9	Rank in top 25%
10	Participating in 1 or more varsity athletic activity, must not be recipient of full athletic scholarship at OU
11	Student Athlete who have earned the varsity letter "O"
12	Pursuing a career in the areas of natural gas, natural gas liquids, other gaseous fuels, and feedstocks
13	Oklahoma high school considered to be small
14	Grandchildren of immigrant who are US citizens, first generation to attend college
15	Non-resident from Midwestern US with special preference to those from IL

Appendix C: Total amount of awards and average amount of awards based on the category of GFN and GPA for full dataset

GFN	GPA	Total Amount of Award			Average Amount of Award		
		One-sided	Two-sided		One-sided	Two-sided	
			Scholarship-oriented	Student-oriented		Scholarship-oriented	Student-oriented
35,001 - 45,000	3-4	211,000	222,000	222,000	1,455	1,520	1,520
	2-2.99	14,000	16,000	16,000	1,400	1,600	1,600
	1-1.99	1,000	1,000	1,000	1,000	1,000	1,000
	0-0.99	32,000	40,000	40,000	1,142	1,428	1,428
25,001 - 35,000	3-4	126,000	102,000	102,000	1,968	1,700	1,700
	2-2.99	4,000	5,000	5,000	1,000	1,000	1,000
	1-1.99	0	0	0	0	0	0
	0-0.99	16,000	16,000	16,000	1,142	1,000	1,000
15,001 - 25,000	3-4	68,000	68,000	68,000	2,833	2,833	2,833
	2-2.99	0	0	0	0	0	0
	1-1.99	0	0	0	0	0	0
	0-0.99	0	0	0	0	0	0
5,001 - 15,000	3-4	62,000	62,000	62,000	2,818	2,818	2,818
	2-2.99	0	0	0	0	0	0
	1-1.99	0	0	0	0	0	0
	0-0.99	0	0	0	0	0	0
0 - 5,000	3-4	15,000	15,000	15,000	3,000	3,000	3,000
	2-2.99	0	0	0	0	0	0
	1-1.99	0	0	0	0	0	0
	0-0.99	0	0	0	0	0	0

Appendix D: Total amount of awards and average amount of awards based on the category of GFN and GPA for sample 1 dataset

GFN	GPA	Total Amount of Award			Average Amount of Award		
		One-sided	Two-sided		One-sided	Two-sided	
			Scholarship-oriented	Student-oriented		Scholarship-oriented	Student-oriented
35,001 - 45,000	3-4	128,000	170,000	170,000	1,523	2,023	2,023
	2-2.99	6,000	6,000	6,000	1,000	1,000	1,000
	1-1.99	0	0	0	0	0	0
	0-0.99	0	0	0	0	0	0
25,001 - 35,000	3-4	115,000	103,000	103,000	1,493	1,337	1,337
	2-2.99	21,000	13,000	13,000	1,615	1,000	1,000
	1-1.99	0	0	0	0	0	0
	0-0.99	0	0	0	0	0	0
15,001 - 25,000	3-4	132,000	102,000	102,000	1,692	1,378	1,378
	2-2.99	11,000	13,000	13,000	1,000	1,000	1,000
	1-1.99	0	0	0	0	0	0
	0-0.99	0	0	0	0	0	0
5,001 - 15,000	3-4	37,000	37,000	37,000	2,846	2,846	2,846
	2-2.99	1,000	0	0	1,000	0	0
	1-1.99	0	0	0	0	0	0
	0-0.99	0	0	0	0	0	0
0 - 5,000	3-4	3,000	3,000	3,000	3,000	3,000	3,000
	2-2.99	0	0	0	0	0	0
	1-1.99	0	0	0	0	0	0
	0-0.99	0	0	0	0	0	0

Appendix E: Total amount of awards and average amount of awards based on the category of GFN and GPA for sample 2 dataset

GFN	GPA	Total Amount of Award			Average Amount of Award		
		One-sided	Two-sided		One-sided	Two-sided	
			Scholarship-oriented	Student-oriented		Scholarship-oriented	Student-oriented
35,001 - 45,000	3-4	124,000	146,000	146,000	1,771	2,085	2,085
	2-2.99	10,000	14,000	14,000	1,666	2,333	2,333
	1-1.99	0	0	0	0	0	0
	0-0.99	0	0	0	0	0	0
25,001 - 35,000	3-4	134,000	130,000	130,000	1,425	1,382	1,382
	2-2.99	2,000	2,000	2,000	1,000	1,000	1,000
	1-1.99	0	0	0	0	0	0
	0-0.99	0	0	0	0	0	0
15,001 - 25,000	3-4	146,000	127,000	127,000	1,586	1,395	1,395
	2-2.99	10,000	10,000	10,000	1,000	1,000	1,000
	1-1.99	0	0	0	0	0	0
	0-0.99	0	0	0	0	0	0
5,001 - 15,000	3-4	59,000	53,000	53,000	2,809	2,523	2,523
	2-2.99	4,000	0	0	2,000	0	0
	1-1.99	0	0	0	0	0	0
	0-0.99	0	0	0	0	0	0
0 - 5,000	3-4	12,000	12,000	12,000	3,000	3,000	3,000
	2-2.99	0	0	0	0	0	0
	1-1.99	0	0	0	0	0	0
	0-0.99	0	0	0	0	0	0

Appendix F: The number of awards given to n-rank students for the full dataset

Student Rank	One-sided	Two-sided	
		Scholarship-oriented	Student-oriented
1	80	16	16
2	1	76	76
3	2		
4	6	2	2
6	1		
7	3	11	11
8	3		
12	3		
13	2	1	1
15	2	1	1
16	1	1	1
17	3	1	1
18	6	2	2
20	1		
21	2		
22	2	2	2
23	1	1	1
24	2	3	3
25	3	1	1
26	6	6	6
27		3	3
28	2	31	31
29	7	5	5
30	4	25	25
32	1	20	20
33	3		
34	2		
35	1		
36	4		
37	2		
40	1		
42	4		
43	1		

Student Rank	One-sided	Two-sided	
		Scholarship-oriented	Student-oriented
44	5		
49	1		
50	3		
51	1	4	
52	1		
53	1		
54	12		
56	1		
57	1		
58	1	7	
59	2		
61	7		
62	2		
63	3		
65	1	4	
66	2		
69	1		4
70	3		
71			7
73	1		
74	1		
76	1		
77	1		
78	5		4
79	3		
81	5		
82	7		
83	2		
85	4		
87	1		
88	2		
89	1	1	1
90	3	1	1

Student Rank	One-sided	Two-sided	
		Scholarship-oriented	Student-oriented
91	3		
92	3	1	1
93	1	2	2
94	1		
95	4		
96	2		
99	2		
100	1		
101	1	1	
102	1	4	1
103	2	1	1
104	2	1	1
106		1	1
107	5	2	2
108	5	1	1
109		1	1
110	1	1	1
111	1	1	1
112		1	1
113	2	1	1
114	1	1	1
115		1	1
116	1		
118	1	1	1
119		1	1
120	1		
122	1		
124	1		
125	4		
126	1		
127		3	3
128	1		19
129			14

Student Rank	One-sided	Two-sided	
		Scholarship-oriented	Student-oriented
130		1	1
131	1		4
132	3		2
134	1		
135	2		
136	2	1	
137	1	2	
138		4	1
139	1		
140	1	1	
141	2	3	1
142	1		
143	1	1	3
144		1	
145	1		
146	1	1	
147		3	1
148	1	9	
149		2	
150		13	
153	1		
158		1	1
159		1	1
161		1	1
168		2	2
169	1		
174		1	1
176		1	3
179		1	2
182		1	
185		2	
187		3	1
191		3	2

Student Rank	One-sided	Two-sided	
		Scholarship-oriented	Student-oriented
193		1	2
194			2
205		1	1
206		1	1
221		1	1
224		1	1
228		1	1
235		1	1

Appendix G: The number of students who received n-rank scholarship for the full dataset

Scholar-ship Rank	One-sided	Two-sided	
		Scholarship-oriented	Student-oriented
1	14	14	14
2	146	146	146
3	50	45	47
4	11	12	7
5	14	18	20
6	16	17	15
7	10	3	9
8	1	6	4
9	4	6	5
10	4	4	3
11	3	5	6
12	4	4	4
13	4	2	2
14	3	4	4
15	1	1	1
16	4	2	2
17	2	5	5
18	1	3	3

Scholar-ship Rank	One-sided	Two-sided	
		Scholarship-oriented	Student-oriented
19	4	1	1
20	3	2	2
21	2	2	2
22	3	2	2
23		1	1
24	2	5	5
25	2	2	2
26	1		1
27	2		1
28	1		1
29	2		1
30		1	
32	1	1	
33		1	
35		1	
38	1		
39	1		

Appendix H: An example of the matching result

Scholarship Account	Scholarship Name	Student ID	Student Name
31786	Scholarship 31786	908	Student 908
31786	Scholarship 31786	006	Student 006
31786	Scholarship 31786	024	Student 024
31786	Scholarship 31786	736	Student 736
31786	Scholarship 31786	737	Student 737
31786	Scholarship 31786	639	Student 639
31786	Scholarship 31786	751	Student 751
31786	Scholarship 31786	758	Student 758
31786	Scholarship 31786	257	Student 257
31786	Scholarship 31786	764	Student 764
31786	Scholarship 31786	597	Student 597
31786	Scholarship 31786	538	Student 538
31786	Scholarship 31786	290	Student 290
31786	Scholarship 31786	845	Student 845
31786	Scholarship 31786	334	Student 334
31786	Scholarship 31786	277	Student 277
31786	Scholarship 31786	215	Student 215
31786	Scholarship 31786	214	Student 214
31786	Scholarship 31786	554	Student 554
31786	Scholarship 31786	039	Student 039
31786	Scholarship 31786	917	Student 917
31786	Scholarship 31786	739	Student 739
31786	Scholarship 31786	579	Student 579
31786	Scholarship 31786	318	Student 318
31786	Scholarship 31786	603	Student 603
31786	Scholarship 31786	761	Student 761
31786	Scholarship 31786	577	Student 577
31786	Scholarship 31786	875	Student 875
31786	Scholarship 31786	139	Student 139
31786	Scholarship 31786	745	Student 745
31786	Scholarship 31786	576	Student 576
31786	Scholarship 31786	100	Student 100
31812	Scholarship 31812	052	Student 052
31812	Scholarship 31812	867	Student 867
31812	Scholarship 31812	141	Student 141
31812	Scholarship 31812	783	Student 783
31812	Scholarship 31812	176	Student 176
31812	Scholarship 31812	584	Student 584