Matching algorithms for the secondary school admission problem in Amsterdam

Wessel Klijnsma
10172432

Bachelor thesis
Credits: 18 EC

Bachelor Opleiding Kunstmatige Intelligentie

University of Amsterdam
Faculty of Science
Science Park 904
1098 XH Amsterdam

Supervisor
Prof. dr. Jan D.J.N van Eijck

Institute for Logic, Language and Computation (ILLC)
Faculty of Science
University of Amsterdam
Science Park 107
1098 XG Amsterdam

June 24, 2016

Abstract

In 2015 there was a controversy among teachers, parents and students about the matching procedure for the secondary school admissions problem. The matching algorithm used in this case was DA-MTB.

The results provided by conducting a series of simulation experiments, with real data and generated data as input, provide insights into the performance and properties of the matching algorithms: DA-MTB, DA-STB, The Boston Mechanism and Random Serial Dictatorship. Outcomes of the algorithms are evaluated on efficiency and fairness, where occurrences of justified envy are used to measure fairness.

The Boston Mechanism maximizes the number of students assigned to their school of first choice. It only produces instances of justified envy if priority rules are enforced for all preferences. DA-MTB assigns the smallest proportion of students to their first choice but maximizes the number of students assigned to a school in their top 3. DA-STB and DA-MTB produce no instances of justified envy. Random Serial Dictatorship produces instances of justified envy when priority rules are enforced for all preferences and when priority rules are only enforced for the first choice.

Contents

Chapter 1

Introduction

Dividing prospective students among the secondary schools of a city is a problem which many municipalities and school boards have to deal with. Especially in cities with a large number of inhabitants this task seems problematic. These problems are caused by the fact that there is usually not enough capacity to assign every students to the school of their first choice. Making compromises that facilitate the allocation of schools is necessary, otherwise no division can be made. It is important to make these compromises in such a way that they are considered fair by the public and provide an allocation where reasonable satisfaction among those concerned in division is achieved.

Matching algorithms are mechanisms that are concerned with the task of matching two groups based on priorities and preferences. In the context of school matching, many-to-one matching algorithms provide a solution for the problem described above. Here, students are matched to a school based on a list preferences of the students and priority rules of the schools. The list of preferences is a list in which students list schools in order of preference. The priority rules provide a way of ordering applicants of a school by priority. For example, a school can give applicants, that have a sibling attending the school, priority over other students.

In 2015, one of these matching algorithms, Deferred Acceptance (MTB), was used to divide students among Amsterdam schools (De Haan et al., 2015b). The choice to use this algorithms was based on a simulation study described by De Haan et al. (2014), in which matching algorithms are compared in terms of their efficiency. The algorithm involved in this study where Deferred Acceptance (DA-MTB), Random Serial Dictatorship and The Boston Mechanism. Here efficiency is defined by the proportion of students that is assigned to a school in the top x of their preference list.

While De Haan et al. (2014) focussed on efficiency, this thesis will also take the fairness of a solution into account. The concept of fairness in this project is based on how algorithms handle justified envy. In the following sections an overview of the literature concerning this topic will be given. Also simulation experiments will be described that investigate the efficiency and fairness of the algorithms.

The matching of 2015 in Amsterdam caused a stir among students, parents and teachers[1]. Many students and parent were not satisfied with the matching procedure due to the fact that about 24 % of the students were not assigned to the school of their first choice (De Haan et al., 2015b).

Besides this issue of not being able to go the school of choice, confusion about matching procedure and feelings of being treated unfair are possible explanations of the discontent among those involved in the matching. The aim of this project is to provide insights about the performance of matching algorithms that can be used by policy makers to make decisions about the algorithms. These insights can also be used to clarify some of the confusion that exists about the matching procedure.

[1] See this news article as reported by newspaper Parool: `http://www.parool.nl/amsterdam/meerderheid-schoolleiders-ontevreden-over-nieuwe-matchingsysteem~a4156727/`

Chapter 2

Theoretical framework

2.1 Literature Review

This section will provide an overview of the literature regarding matching algorithms. Furthermore, a conceptual analysis of the algorithms and its properties are presented. A detailed description of the algorithms and the definition of the properties will be given in subsequent sections.

The Deferred Acceptance matching algorithm was proposed by Gale and Shapley (1962) in their famous article *College admissions and the stability of marriage*. By extending the one-to-one matching algorithm for stable marriage to the many-to-one algorithms for college admissions, Deferred Acceptance has proven to be versatile. The matching is done on the basis of preferences of both parties. It has been applied to solve matching problems in several domains, ranging from the housing market to college admissions .

Gale and Shapley (1962) prove that Deferred Acceptance provide matchings that are both **stable** and **optimal** when applied to the college admissions problems. Furthermore, Roth (2008) proposes that the matchings are **pareto efficient** and that the algorithm is **strategy proof**. Abdulkadiroglu and Sönmez (2003) describe how Deferred Acceptance can be applied to the secondary school admissions problem. They identify the fact that in the secondary schools admissions problem only students have preferences to be the main difference with the college admissions problems where the preferences are two sided. Despite this difference, Abdulkadiroglu and Sönmez (2003) show that Deferred Acceptance is still useful in the design of a matching mechanism for school assignment due to its stable properties. Abdulkadiroglu and Sönmez (2003) also show that Deferred Acceptance eliminate **justified envy**

In a report by De Haan et al. (2015b) two versions of the Deferred Acceptance algorithm for school assignment are discussed. One version uses a single centralized lottery to break ties in priority, this version is known as Deferred Acceptance Single Tie Breaking (DA-STB). The other version uses a lottery at each school to break ties, this version is known as Deferred Acceptance Multiple Tie Breaking (DA-MTB).

Another matching algorithm for the school admission problem is the Boston mechanism. Similar to Deferred Acceptance, students are assigned to school based on their preferences. It uses a lottery to order students in the same priority group (Abdulkadiroğlu et al., 2005). Regardless of the fact that this algorithm provides **pareto efficient** matchings, it has the property that it is not **strategy proof** (Abdulkadiroglu and Sönmez, 2003).

The last algorithm to be discussed in this section is Random Serial Dictatorship. Abdulkadiroglu and Sönmez (2003) describe that algorithm. Again, the matching is performed on the basis of the preferences of the students. A single lottery determines the order in which students are assigned to their most preferred school that has available seats. Abdulkadiroglu and Sönmez (2003) propose that this algorithm is both **strategy proof** and provides **pareto efficient** matchings.

De Haan et al. (2014) report a simulation study applied to Amsterdam schools. In this study, Random Serial Dictatorship, The Boston Mechanism and DA-MTB were simulated using data from questionnaire in which Amsterdam school students were asked about their school preferences. Using Random Serial Dictatorship 90% of the students were given their first choice, 86.4% for The Boston Mechanism and 82.5% for DA-MTB. Using another measure, the percentage of students that were assigned to a school of their top 3, the results were 99.9% for DA-MTB, 98.1% for random serial dictatorship and 96.1% for the Boston mechanism.

De Haan et al. (2015a) matching process in Amsterdam. In 2015 an implementation of the Deferred Acceptance (DA-MTB) algorithm was used for the assignment of students to secondary schools. As a result, 74% of the theoretical 87% were given first choice.

While De Haan et al. (2014) focussed only on efficiency, Morrill (2015a) identifies another measure in which the performance of the algorithms can be evaluated. This measure is concerned with the fairness. Morrill (2015a) describes how fairness can be determined by investigating instances of **justified envy**.

2.2 Research question

Answering the following research question will provide valuable insights about the qualities of the algorithms. These insights can be used by policy makers to make a well informed decision about which matching algorithms to use. The research question is: *Out of the matching algorithms DA-STB, DA-MTB, Random Serial Dictatorship and the Boston Mechanism, which one performs best, in terms of **efficiency** and **fairness**, in the context of Amsterdam secondary schools and students?*

2.3 Definitions

For clarification, this section will provide definitions of several concepts concerning matching algorithms.

Definition 2.3.1. (Gale and Shapley, 1962), "An assignment of applicants to colleges will be called **unstable** if there are two applicant α and β who are assigned to colleges A and B, respectively, although α prefers B and β prefers A." (p. 3)

Definition 2.3.2. (Gale and Shapley, 1962), "A stable assignment is called **optimal** if every applicant is as least as well off under it as under any other stable assignment." (p. 3)

Definition 2.3.3. An outcome of a matching algorithm is **pareto efficient** if it is impossible to improve the assignment of a student without making making the assignment of another student worse.

Definition 2.3.4. A matching algorithm is **strategy proof** if it impossible for a student to gain welfare by being dishonest about his or her preferences.

Definition 2.3.5. (Morrill, 2015a), "Student i is said to have **justified envy** if there is a school a such that i prefers a to her assignment, and i has higher priority[1] at a than one of the students assigned to s" (p. 10)

2.4 Algorithms

In order to fully grasp the workings of the algorithms, this section will provide detailed descriptions of the algorithms. The flowcharts used in this description also serve a reference of how the algorithms were implemented.

2.4.1 Deferred Acceptance

The Algorithms 1 and 2 (in the Appendix) show the pseudocode of the DA-STB and DA-MTB algorithms respectively. For further clarification, one can consult the Flowchart depicted in Figures A.1 and A.2.The difference between the two algorithms can be found by looking at the *Draw priority numbers* step. This step determines the order in which students of the same priority group are assigned. In the DA-STB process this takes at the beginning of the procedure for all the students at the same time, which is known a centralized lottery. For DA-STB the lottery takes place for each school individually when a school considers proposals of students.

[1] Here having a higher priority means that student i has priority over the other student because of a priority rule

2.4.2 The Boston Mechanism

The pseudocode of the Boston Mechanism is shown in Algorithm 3. The main difference with the Deferred Acceptance algorithms is that once The Boston Mechanism has accepted proposals, these are definite and not reconsidered in subsequent steps.

2.4.3 Random Serial Dictatorship

In Random Serial Dictatorship algorithms students are first ordered according to a centrally drawn lottery. Following this order of the students, each students is assigned to the school that still has available seats. If a student has priority at a school but this school has no remaining seats, the student ranking lowest in the priority order is removed. The process then returns to this removed student and assigns it no the next school his or her priority list. Pseudocode for this algorithms is shown in 4. Also, the flowchart in Figure A.4 show the workings of the algorithm.

Chapter 3

Method

In order to answer the research question, the components fairness and efficiency of the performance of the algorithms need to be examined. The method for examining these two components is to use simulation experiments. In the simulation experiments, implementations of the algorithms are run with either generated or real data as input. The output of the algorithms is a matching that can be evaluated in various ways depending on the metric. This section will describe how the algorithms were implemented. Also evaluation metrics will be defined. Lastly, details of the experiments are specified.

3.1 Implementing the algorithms

For this project, the algorithms were implemented in the Python 3[1] scripting language. In addition, the packages Scipy[2], Numpy[3] and Matplotlib[4] were used for generating data, evaluating outcomes and plotting results, respectively. The reason for choosing this language is that the tools needed for the implementation of the algorithms can be accessed in a straightforward manner. The speed of the implementations is irrelevant for this research as only the outcomes are examined.

The algorithms were implemented according to the pseudocode shown in Algorithms 1, 2, 3 and 4. The format of the input and output are the same of all of the three algorithms.

[1]Available at: `https://www.python.org/download/releases/3.0/`

[2]Available at `https://www.scipy.org/`

[3]Available at `http://www.numpy.org/`

[4]Available at `http://matplotlib.org/`

3.2 Evaluation

As mentioned before the performance of the algorithms will be measured in terms of efficiency and fairness. This section will define metrics to measure these two components.

Efficiency

The efficiency of an outcome of an algorithm is information about the capability of an algorithms to assign students to a school as high as possible on their preference list. This information can be found by looking at the distribution of the position of the assigned school on the preference list of the students.

Fairness

For this project fairness will be measured in terms of the number of occurrences of justified envy. Justified envy is defined in Definition 2.3.5. The motivation behind using this measure is that experiencing justified envy as a student can cause a justified feeling of being treated unfair because his or her right of priority has been ignored.

3.3 Experiment I: generated data

The first experiment is conducted with the aim to examine the sensitivity of the efficiency of the algorithms to many versions of the input data. This will provide insight into the overall theoretical efficiency. It is theoretical because of the fact that the generated data is not real data and will therefore fail to capture the nuances present in the in the input data of real school matchings. The evaluation of fairness is disregarded in this experiment because the theoretical properties about the elimination of justified envy are already known. In experiments IIa and IIb this will be taken into account.

The flowchart in Figure 3.2 shows how this experiment was conducted. First, data about the preferences of the students are generated. Subsequently, these data are used as input of the algorithms. After the execution of the algorithms, the outcome of each algorithm is evaluated according to the efficiency metric defined in Section 3.2. This results is then saved. The proces of generating data, running the algorithms and collecting results is repeated t number of times.

The data is generated by initializing n and m schools and students. The preferences of students are picked based on a randomized discrete probability distribution that determines how likely it is for a school to be picked. This distribution can be used to draw numbers that correspond to a school ID. Using this distribution a set of unique numbers can be drawn. This set is the preference list of a student. In an attempt to make the data more realistic, the students are divided in to preference groups. Students in the same preference group are likely to have similar preferences. Their preferences are picked according to the same distribution. The division of students into preference groups will simulate the effect that students with similar background, influenced by various factors e.g: neighborhood, friends, primary school, are likely to have similar preference lists.

For this experiment, values $n = 10$, $m = 300$ and $t = 1000$ are used. Meaning that 300 students are divided among 10 school a 100 times. After collecting results, they are analyzed by averaging the number of students that where assigned to a school in there top x, with x ranging from 1 till 10.

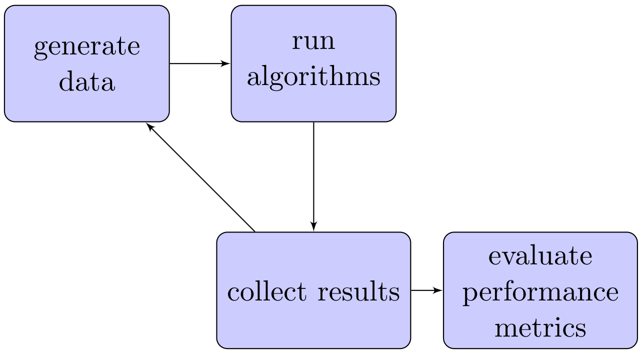

Figure 3.1: Flow chart of experiment I

3.4 Experiment IIa: real data

In experiment IIa real data are used to test the efficiency and fairness of the algorithms. The data that are used are about the 2015 matching in Amsterdam provided by OSVO[5]. It lists the students and their preferences and schools and their capacity. Also for each school in the preference list a boolean value is provided whether or not a student has priority at that school.

Unlike the data used in experiment I, this data originates from the real world and is therefore realistic. Despite the fact that the theoretical properties about the elimination of justified envy of the algorithms are known, this experiment will evaluate the fairness of outcomes to test the fairness of the algorithms with real input data. Examining the efficiency and fairness using this data can be used to obtain insight into the performance of the

[5]OSVO is an association of all the school boards of primary and secondary schools in Amsterdam. OSVO is responsible for carrying out the matching. For more information see `http://www.verenigingosvo.nl/`

algorithms in a real life situation.

The data consist of 2 Excel files, one containing the capacity of each school and one containing the preference lists of the students. Before using the data for the experiment several preprocessing steps had to be taken. First, the Excel files were converted to Comma Separated Values files. Then the names of the schools in the preference lists were substituted with an ID number. The newly created CSV files are then read in to the program that handles the experiments and converted to the input format of the algorithms.

In the 2015 Amsterdam matching 7510 students were divided among 136 schools. In this experiment, the algorithms are run with this input data a 100 times. Each time statistics about the efficiency and occurrences of justified envy are collected. In the end these statistics are averaged. For this experiment, priorities are enforced for each position on the preference list.

3.5 Experiment IIb: real data

In this variation of experiment IIa, priorities are only enforced for schools listed first on preference lists. The matching mechanism used in Amsterdam in 2015 uses the same rule; priorities only count for the schools of first choice. Besides changing the rule about when priority rules are enforced, the procedure is the same as in experiment IIa.

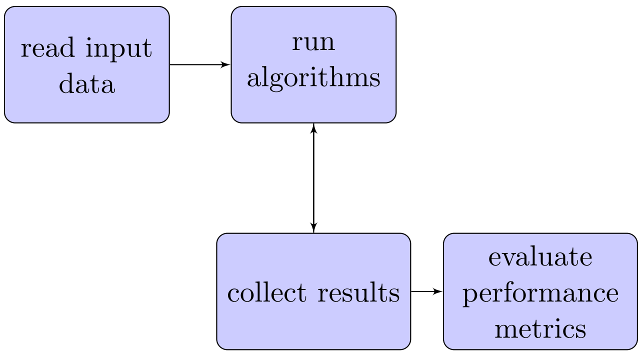

Figure 3.2: Flow chart of experiment II

Chapter 4

Results

Figures 4.1, 4.2 and 4.3 show the cumulative share of the proportion of the rank of the assigned school on the preference list, for experiments I, IIa and IIb respectively. Occurrences of justified of experiments IIa and IIb are listed in Table 4.1 and 4.2, respectively.

4.1 Experiment I

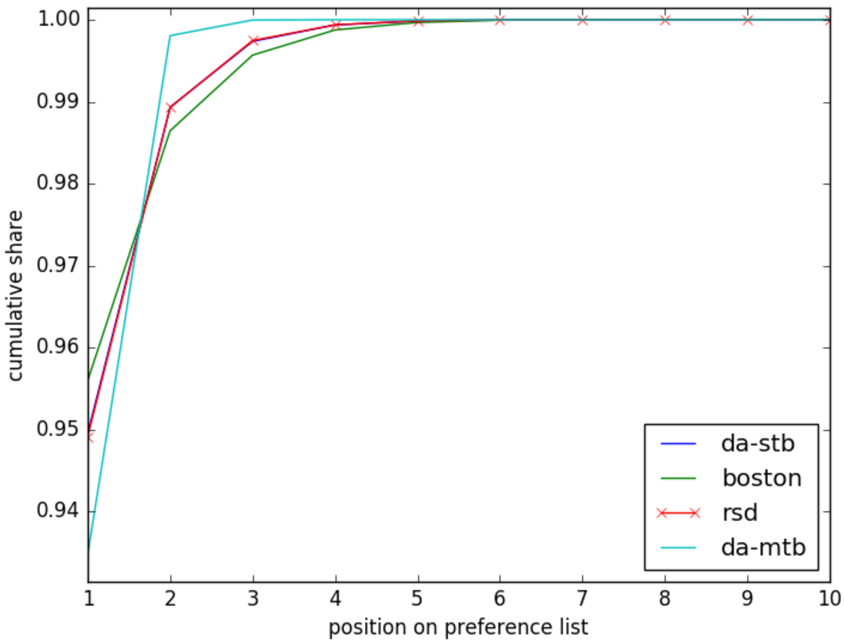

Figure 4.1: Result efficiency of experiment I

4.2 Experiment IIa

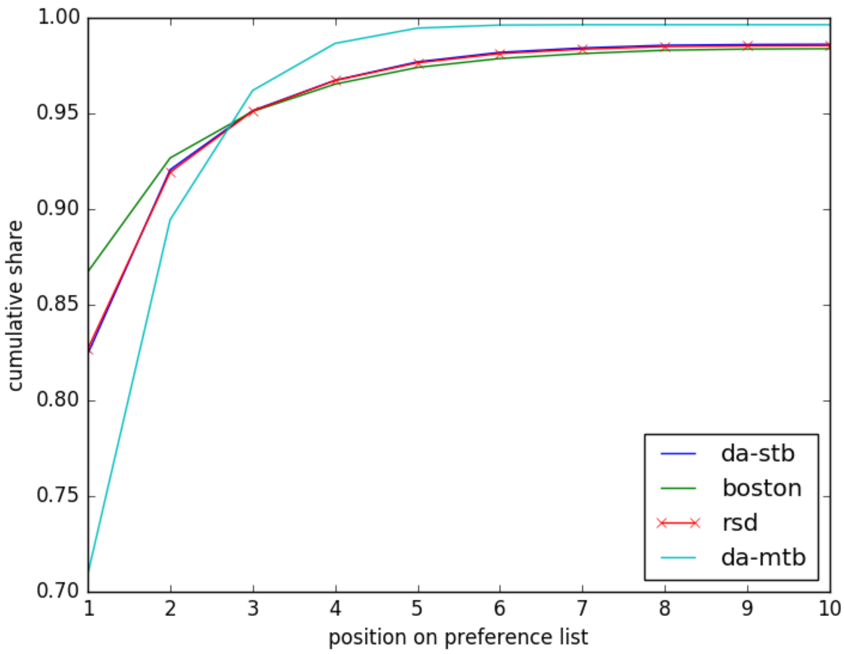

Figure 4.2: Result efficiency of experiment IIa

Table 4.1: Occurrences of justified envy in experiment IIa

	average occurrences of justified envy	percentage of total students
DA-STB	0	0
DA-MTB	0	0
The Boston Mechanism	11.94	0.00159
Random Serial Dictatorship	38.95	0.00519

4.3 Experiment IIb

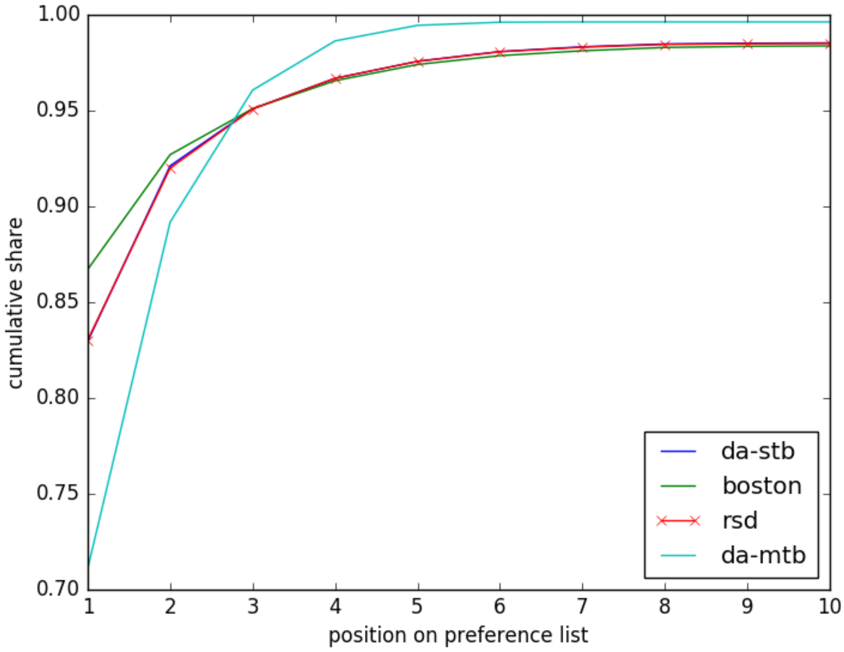

Figure 4.3: Result efficiency of experiment IIb

Table 4.2: Occurrences of justified envy in experiment IIb

	average occurrences of justified envy	percentage of total students
DA-STB	0	0
DA-MTB	0	0
The Boston Mechanism	0	0
Random Serial Dictatorship	28.74	0.00382

4.4 Analysis

Looking at the results displayed in the previous sections, there are several observations regarding the efficiency that can be made. Firstly, for all the experiments DA-MTB assigns the lowest number of students to their school of choice. However, in experiment I, DA-MTB assigns more students to a school in their top 2 and higher than the other algorithms. The same effect occurs in experiments IIa and IIb at top 3 and higher. Also, DA-MTB assigns more students to a school on their preference list than the other algorithms. Secondly, The Boston mechanism assigns the highest number of students to the school of choice in experiments I, IIa and IIb. In experiments IIa and IIb it also has the largest proportion of students assigned to a school in their top 2. Lastly, in all the experiments, RSD and DA-STB have similar efficiency. The number of students that is assigned to the first choice is higher than DA-MTB but lower than The Boston Mechanism. This also counts for top 2 in experiments IIa and IIb.

The following observations about the ability of the algorithms to avoid occurrences of justified envy can be made. In accordance with the fact that Deferred Acceptance algorithms eliminate justified envy, DA-MTB and DA-STB produce allocations that are free of occurrences of justified envy in experiments IIa and IIb. The Boston mechanism produces allocations that contain instances of justified envy in experiment IIa. In experiment IIb, the allocations produced by the Boston Mechanism are free of justified envy. The allocations produced by RSD are not free of justified envy in both experiment IIa and IIb. The number of instances in IIa is higher than the number of instances in IIb. Also, in experiment IIa, the number of instances produced by RSD is higher than produced by The Boston Mechanism. For both RSD and The Boston mechanism, the percentages of instances of justified envy is small and can therefore occurrences of justified envy can be considered as rare.

Chapter 5

Discussion

Taking the results and the analysis of the results into account, it is hard to determine which algorithms performs best. This is due to the fact that the distinction between 'good' and 'bad' are based on subjective norms.

For example, if one prefers to maximize the number of the students assigned to the school of their first choice regardless of the small proportion of students assigned to school low on their list, the Boston mechanism should be chosen. If there is a requirement for a strategy proof, DA-STB or RSD should be chosen in this scenario. A possible explanation of the fact there was discontent about the 2015 Amsterdam matching is that being assigned to the school of second or third choice is disappointing. This scenario should be chosen if one wants to avoid this disappointment.

Another scenario would be to prefer to maximize the number of students assigned to school in their top 3, thereby minimizing the number of students being given a school not in their top 3. In this scenario, in which the 'load is spread' across all of the students, the DA-MTB algorithms would be the best algorithm.

Besides choosing between the scenarios described above, there is the question whether lack strategy proofness is bad. Again, this is a matter of opinion. Perhaps the unfair advantage of the students from well educated backgrounds can be avoided by educating parents and students of poorly educated backgrounds.

Lastly, it could be considered whether the fact that an algorithm produces a small number of instances of justified envy is a reason to not choose this algorithm. If so, RSD should not be chosen. And the Boston mechanism should not be chosen if priority rules are enforced for all preferences.

The insights gained from this research can be used by policy makers to choose the right algorithm. In order to do so, the questions posed above have be answered. Besides using the gained insigths, policy makers also have to take the public opinion regarding these matters.

5.1 Future Work

In this section two recommendations for future research will be made. Firstly, since the choice for the 'right' algorithm depends on public opinion and politics, it would a good idea to ask the public about their views on the topics discussed in the previous section. Perhaps by reevaluating the performance of the algorithms based metric that were established using questionnaires will cause less discontent among teachers, parents and students.

Another suggestion is to include the Top Trading Cycles algorithm in the comparison. Abdulkadiroglu and Sönmez (2003) show that this algorithm can successfully be used for the secondary school admission problem. The Clinch and Trade variations, as described by Morrill (2015b), may also be worthwhile considering.

Appendices

Appendix A

Algorithms

A.1 Pseudocode

Algorithm 1 Pseudocode for the DA-STB algorithm

```
function DaStbMatching(Schools, Students)
    Matching := List( )
    Order := DrawOrder(Students)
    NotPlacable := List( )
    while StudentsLeftToAssign(Students) do
        MatchingRound(Students, Schools, Order, Matching, NotPlacable)
    end while
    RandomAssignment(NotPlacable, Schools, Matching)
    return Matching
end function

function MatchingRound(Students, Schools, Order, Matching, NotPlacable)
    for S ← Students do
        List := SelectPreferences(S)
        if SchoolLeftOnList(List) then
            ProposeNextFavorite(List, S, Schools)
        else
            Add(S, NotPlacable)
        end if
    end for
    for School ← Schools do
        Proposals := GetProposals(Schools)          ▷ these include previously accepted
        SortProposalsToOrderAndPriority(Proposals, Students, Order)
        AcceptMaxNoOfProposals(Proposals, Students, School, Matching)
    end for
end function
```

Algorithm 2 Pseudocode for the DA-MTB algorithm

```
function DaMtbMatching(Schools, Students)
    Matching := List( )
    NotPlacable := List( )
    while StudentsLeftToAssign(Students) do
        MatchingRound(Students, Schools, Matching, NotPlacable)
    end while
    RandomAssignment(NotPlacable, Schools, Matching)
    return Matching
end function

function MatchingRound(Students, Schools, Matching, NotPlacable)
    for S ← Students do
        List := SelectPreferences(S)
        if SchoolLeftOnList(List) then
            ProposeNextFavorite(List, S, Schools)
        else
            Add(S, NotPlacable)
        end if
    end for
    for School ← Schools do
        Proposals := GetProposals(Schools)        ▷ these include previously accepted
        Order := DrawOrder(Students)
        SortProposalsToOrderAndPriority(Proposals, Students, Order)
        AcceptMaxNoOfProposals(Proposals, Students, School, Matching)
    end for
end function
```

Algorithm 3 Pseudocode for the Boston algorithm

```
function BostonMatching(Schools, Students)
    Matching := List( )
    NotPlacable := List( )
    Order := DrawOrder(Students)
    P := 0
    while StudentsLeftToAssign(Students) do
        Unplacable := SelectUnplacableStudents(P, Students)    ▷ these students have no preferences left on list
        Add(Unplacable, NotPlacable)
        for School ← Schools do
            Applicants := SelectStudentsWithSchoolAtRank(P, School, Student)
            SortApplicantsToOrderAndPriority(Applicants, Students, Order)
            AcceptMaxNoOfProposals(Applicants, Students, School, Matching)
        end for
        P := P + 1
    end while
    RandomAssignment(NotPlacable, Schools, Matching)
    return Matching
end function
```

Algorithm 4 Pseudocode for the Random Serial Dictatorship algorithm

```
function RsdMatching(Schools, Students)
    Shuffle(Students)
    NotPlacable := List( )
    Matching := List( )
    for S ← Students do
        P := 0
        Unplacable := False
        List := SelectPreferenceList(S, Students)
        School := SelectSchoolOfList(List, Schools, P)
        while ! SeatsLeft(School) do
            P := P + 1
            List := SelectPreferenceList(S, Students)
            School := SelectSchoolOfList(List, Schools, P)
            if P ≥ Length(List) then
                Unplaceable := True
                Add(S, NotPlacable)
                break
            end if
            if HasPriority(S, School) then
                RemovedStudent := RemoveLowestPriority(School)
                Reconsider(RemovedStudent)
                break
            end if
        end while
        if ! Unplacable then
            Assign(S, School, Matching)
        end if
    end for
    RandomAssignment(NotPlacable, Schools, Matching)
    return Matching
end function
```

A.2 Flowcharts

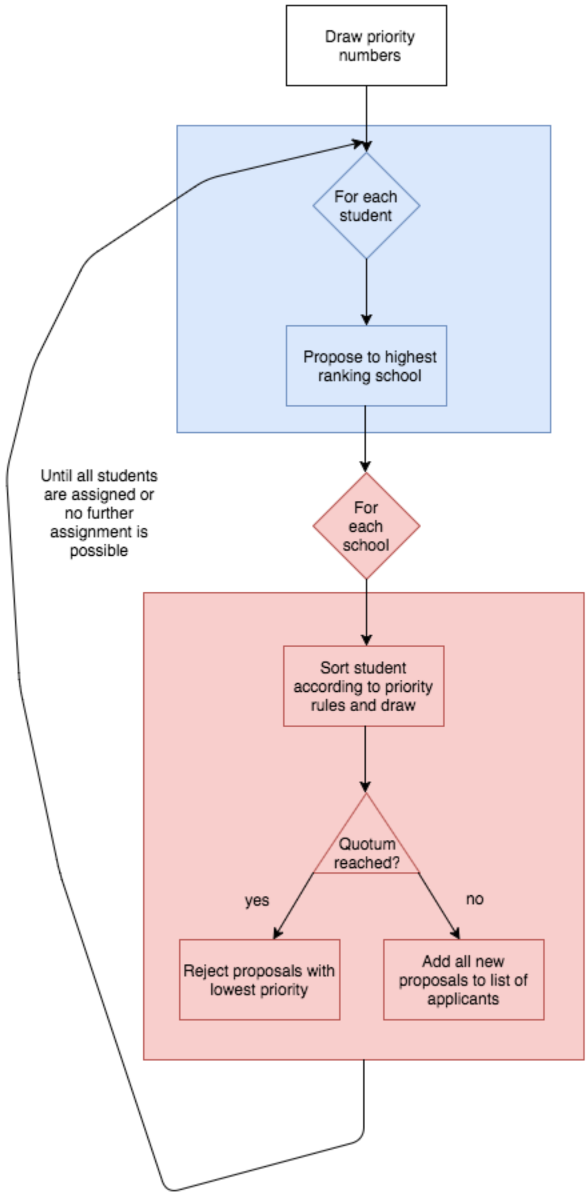

Figure A.1: Flowchart of the DA-STB algorithm

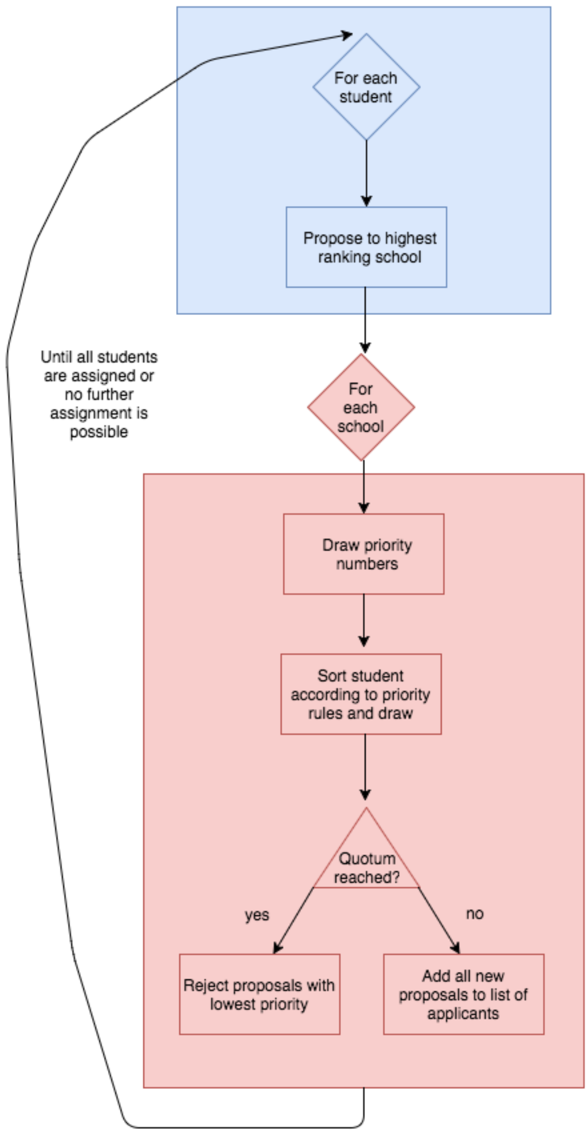

Figure A.2: Flowchart of the DA-MTB algorithm

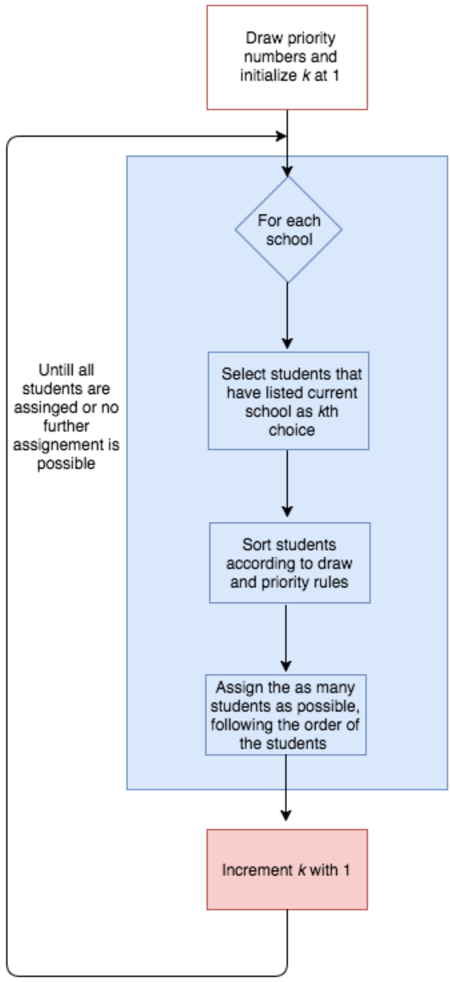

Figure A.3: Flowchart of the Boston Mechanism algorithm

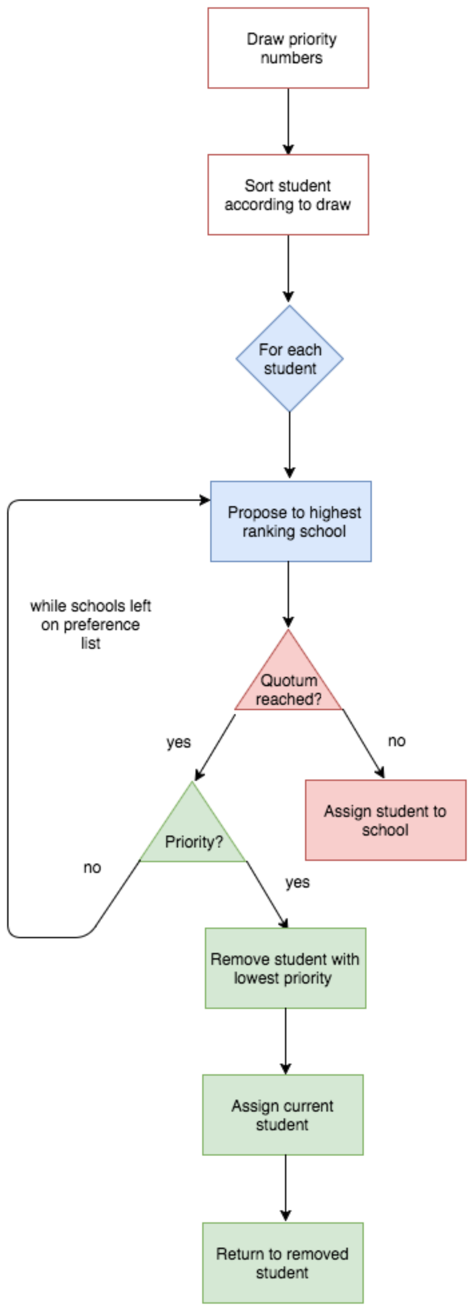

Figure A.4: Flowchart of the Random Serial Dictatorship algorithm

Bibliography

Abdulkadiroğlu, A., Pathak, P. A., Roth, A. E., and Sönmez, T. (2005). The boston public school match. *American Economic Review*, pages 368–371.

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

De Haan, M., Gautier, P. A., Oosterbeek, H., and Van der Klaauw, B. (2014). Schoolkeuze voorgezet onderwijs in amsterdam: Verslag van een simulatiestudie. `http://www.verenigingosvo.nl/wp-content/uploads/2014/04/RapportSimulaties.pdf`. [Online; accessed 12-04-2016], in Dutch.

De Haan, M., Gautier, P. A., Oosterbeek, H., and Van der Klaauw, B. (2015a). Eerste analyse matching en loting voortgezet onderwijs amsterdam 2015. `http://www.verenigingosvo.nl/wp-content/uploads/2015/07/Evaluatie_Matching_beschrijvend_rapport.pdf`. [Online; accessed 12-04-2016], in Dutch.

De Haan, M., Gautier, P. A., Oosterbeek, H., and Van der Klaauw, B. (2015b). The performance of school assignment mechanisms in practice.

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

Morrill, T. (2015a). Making just school assignments. *Games and Economic Behavior*, 92:18–27.

Morrill, T. (2015b). Two simple variations of top trading cycles. *Economic Theory*, 60(1):123–140.

Roth, A. E. (2008). Deferred acceptance algorithms: History, theory, practice, and open questions. *International Journal of game Theory*, 36(3-4):537–569.