Treball final de grau

GRAU DE MATEMÀTIQUES

Facultat de Matemàtiques
Universitat de Barcelona

Two-sided Matching Theory

Autor: Helena Fàbregas Vàzquez

Director: Dra. Marina Núnez Oliva
Realitzat a: Departament de Matemàtica Econòmica, Financera i Actuarial.

Barcelona, June 29, 2015

Abstract

The purpose of this degree project is to study two-sided matchings where money is not involved.

Matching theory is a branch of discrete mathematics belonging to game theory. This theory considers markets with two disjoint sets, such as men and women, firms and workers or colleges and students. Each agent on one sector has preferences (a complete and transitive binary relation) over the set of agents on the opposite side. Then, a matching is a set of pairs formed by agents of different side, in such a way that one agent can take part in at most one pair.

We can situate its origin in the article of Gale and Shapley (1962) "College admissions and the stability of marriage" followed by the book of Knuth (1976), which first edition in French had the title of "Mariages stables".

The first chapter of this monograph focuses on the theory of one-to-one matching, that is known as the marriage problem. This chapter provides the theoretical basis to develop two-sided matching theory, since the notions of stability and optimality for matchings are studied in depth. Chapter 2 is devoted to many-to-one matching problems, say the college admission problem, to analyse until which extent the results obtained for one-to-one markets still hold. In these two chapters the existence of stable matchings, their properties and the structure of the set of stable matchings are studied.

Chapter 3 is a real-life application of the theory of matchings: the school choice problem. Here, we are going to analyse which algorithms have been used to fairly assign children to schools. This problem is currently under study, approached from the fields of mathematics, economics, operations research or computer science.

Contents

Introduction

The theory of matching is a branch of (discrete) mathematics that from its very beginning attracted the interest of economists. We can situate its origin in the article of Gale and Shapley (1962) "College admissions and the stability of marriage" followed by the book of Knuth (1976), which first edition in French had the title of "Mariages stables".

This theory considers markets with two disjoint sectors (two-sided markets), such as men and women, firms and workers or colleges and students. Each agent on one sector has preferences (a complete and transitive binary relation) over the set of agents on the opposite side. Then, a matching is a set of pairs formed by agents of different side, in such a way that one agent can take part in at most one pair. The first desired requirement for a matching is stability: a matching is stable if no pair of agents prefer to break their respective partnerships in order to be matched together.

Gale and Shapley (1962) provide and algorithm, the deferred acceptance algorithm, to obtain a stable matching for any such two-sided market. Moreover, this matching is proved to be optimal (among all other stable matchings) for agents on one of the sides of the market, while it is the worst stable matching for all agents of the other side. Although they formalize the problem in terms of a marriage market (as a good example of a one-to-one market) the real economic motivation is the college admission problem, in which each student is assigned to only one college, but each college can receive several students, in fact as many students as a given quota or capacity attached to the college. This is a first example of market design, since they prove that the same deferred acceptance algorithm provides a stable matching in this many-to-one market.

In the years that followed, mathematicians worked on the marriage problem, basically studying the lattice structure of the set of stable matchings (Knuth, 1976) and providing algorithms to describe the full set of stable matchings (Gusfield and Irving, 1989). Also in these years, researchers noticed that the two-sided characteristic of the market was crucial for the existence of a stable matching, since there are markets with its agents organized either in only one sector or in three disjoint sectors in which no stable set of pairs or triplets exists.

Another important feature of the marriage problem and the college admission problem is that there is no money in these markets: partners cannot be obtained by paying a price. In comparison to that, Shapley and Shubik (1972) present a companion model, "the assignment game", where there are two disjoint sets of agents, say buyers and sellers, and a valuation matrix that gathers how much each buyer values the object that each seller has on sale. The outcome in this markets is not only a matching but also a vector of prices, the amount that each buyer pays to his assigned seller.

The present monograph focuses on the two-sided matching markets without money, since they have proved to have interesting theoretical properties and moreover have been successfully used in the last years for the design of mechanisms for real-life market situations.

In the 80's of the past century, Alvin Roth was asked to redesign the mechanism to allocate medical students to hospitals, what was known as the National Resident Matching Program (NRMP). He discovered that in the early 50's, the NRMP had independently discovered Gale and Shapley's deferred acceptance algorithm with hospitals proposing and was still using it. Roth (1984) proposes to change to the deferred acceptance algorithm with students proposing, in order to obtain the best matching for students and also because it guarantees that students will report their true preferences. Later on, this mechanism was slightly modified to take into account several facts, like for instance the presence of couples of young doctors willing to be allocated to near hospitals (Roth and Peranson, 1999).

The above NRMP is an example of college admission problem. For this model, additional notions of stability, such as group stability, can be considered. To define group stability, colleges need to be able to compare not only individual students but also groups of students, to determine whether one matching is better than another for a college. For arbitrary preferences, group-stable matchings may not exist. But if preferences of colleges over sets of students are responsive to the preferences they have over individuals, then pairwise stability coincides with group stability and existence is hence guaranteed.

School choice problems are another example of market design. In a school choice problems, students have preferences over schools and again each school has a given quota or capacity. The difference is that now schools do not have preferences over children, just a priority order over them that comes from objective issues determined by the school authorities: living in walking distance to school, having siblings in the same school... This fact makes this market not a real two-sided market. However, the notion of stable matching can be similarly defined.

Each school district has a different procedure to solve this allocation problem. In Boston the mechanism used before 2005 was not satisfactory for the authorities since it was not envy-free (a student could be allocated to a school he did not liked much, while another school he preferred more had accepted another student with lower priority than himself). The mechanism was also not strategy-proof, it gave incentives to parents to misrepresent their true preferences over schools. Abdulkadiroglu, Pathak, Roth and Sonmez (2005) proposed to change either to Gale and Shapley's Deferred Acceptance mechanism or to Gale's Top Trading Cycle mechanism. The first one produces a stable matching and is strategy proof, but may fail to achieve efficiency. On the other side, the Top Trading Cycle mechanism is efficient and strategy-proof, but may fail to satisfy stability. Finally, the Deferred Acceptance mechanism was adopted in Boston school district, and also in New York High School admission problem (Abdulkadiroglu, Pathak and Roth, 2005).

The first chapter of this monograph focuses on the theory of one-to-one matching, that is known as the marriage problem. Chapter 2 is devoted to many-to-one matching problems, say the college admission problem, to analyse until which extent the results obtained for one-to-one markets still hold. In these two chapters the existence of stable matchings, their properties and the structure of the set of stable matchings are analysed. Together with the articles where this research was initially published, our monograph relies on the book of Roth and Sotomayor (1990), "Two sided matching: a game-theoretical approach", the book of Knuth (1976), "Stable marriage and its relation to other combinatorial problems", and the text of Gusfield and Irving (1989), "The stable marriage problem: structure and algorithms". The reader will realize from the literature that, from the very beginning, this kind of problems have been approached from the fields of mathematics, economics, operations research and computer science.

Two real-life applications of the college admission problem are included in this monograph. Chapter 2 includes the National Resident Matching Program and the School Choice Problem is developed in Chapter 3.

There exist other interesting applications of the theory of matching to the Roomate Problem, the Housing Allocation and Exchange and the Kidney Exchange. For a survey on these topics see the paper of Alvin Roth in Econometrica (2002), his Hahn Lecture (2008) or his Noble Lecture (2012). Also Sonmez and Unver chapter in the Handbook in Social Economics (2011).

In all this theory, and also in its practical applications, two algorithms are essential: the Deferred Acceptance Algorithm and the Top Trading Cycle Algorithm. We include their codes in an appendix.

David Gale, who was one of the authors of both algorithms, died in 2008. The Sveriges Riksbank Prize in Economic Sciences in Memory of Alfred Nobel 2012 was awarded jointly to Alvin E. Roth and Lloyd S. Shapley "for the theory of stable allocations and the practice of market design".

Chapter 1

The marriage model

The marriage problem is a particular case of a two-sided matching market. In a two-sided market, the agents are partitioned in two disjoint sets. In contrast with other two-sided models, between buyers and sellers or firms and workers, in which utility is transferable by means of prices or salaries, the data of these markets are just the preferences of each agent over the agents on the opposite side. The output, of such a market, is a matching, that is, a partition of the agents in pairs formed by agents of different sector or individuals.

Typically, in the literature, this market, is known as a marriage market, since, as in marriage, each agent is matched with only one agent of the opposite side.

In chapter 2 we will consider similar markets where some agents may have several partners, and hence the image of marriage will not be useful any more.

1.1 Stable matchings

The aim of this chapter is to present formally the marriage model, focusing on finding matchings that nobody will regret, that is, stable matchings, and deeply study their properties.

1.1.1 The formal model

As stated before, the purpose of this section is to introduce terminology and notation about marriage markets and the main concepts of a formal cooperative model.

We will suppose the general rules governing this market are these:

1. Any man and woman who both consent to marry one to another may proceed to do so.

2. Any man or woman is free to withhold his or her consent and remain single.

Let M and W be two finite and disjoints sets, $M = \{m_1, \cdots, m_n\}$ called the set of the men and and $W = \{w_1, \cdots, w_p\}$ called the set of the women. Each man $m \in M$ has preferences over the women and each woman $w \in W$ has preferences over the men.

To express these preferences concisely, the preference of each man m will be represented by an ordered list of preferences, $P(m) \in W \cup \{m\}$. A man m's preferences might be of the form:

$$P(m) = \{w_3, w_4, m, w_1, \cdots, w_p\}.$$

The first choice of the man m is to be married whit w_3, the second is to be married with w_4 and his third choice is to remain single.

If a man m' is indifferent between several options, say w_2, w_3, w_5, we denote them by brackets in his list:

$$P(m') = \{w_1, [w_2, w_3, w_5], m', \cdots, w_p\}.$$

Similarly, each woman $w \in W$ has an ordered list of preferences, $P(w) \in M \cup \{w\}$.

Since each man (woman) will never match a woman (man) that is after the option of remaining single in his preference list, we will usually describe agent's preferences by writing only the ordered set of people that are preferred more than being single. Hence, in (1.1.1) we write:

$$P(m) = \{w_3, w_4\}.$$

Definition 1.1.1. *We denote a specific marriage market by the triple (M, W, P), where:*

- *M the men's set.*
- *W the women's set.*
- *P is the set of preferences lists $P = \{P(m_1), \cdots, P(m_n), P(w_1), \cdots, P(w_p)\}$.*

Following with notations and terminology, we write $w >_m w'$ to mean the man m prefers the woman w more than woman w', and $w \geq_m w'$ to mean m prefers w at least as much as w'. Moreover, woman w is acceptable to man m if he likes her at least as much as remaining single, that is, if $w \geq_m m$.

Formally, a preference of an individual is a binary relation over its set of alternatives that is complete (any two alternatives can be compared) and transitive. This is the reason we can represent it by a list.

An individual player has strict preferences if he or she is not indifferent between any two acceptable alternatives.

Definition 1.1.2. *We say that individuals have rational preferences if their preferences have the property of transitivity and complete ordering.*

Our goal is to find out what kind of outcome will result from the collective interaction between men and women. An outcome of the marriage market is a set of marriages. In general, and due to the rules of the game, some people may remain single, i.e. self-matched. Formally we define:

Definition 1.1.3. *A matching, of the marriage market, is a one to one correspondence* $\mu : M \cup W \longrightarrow M \cup W$*, such that,*

- $\mu(m) \in W \cup \{m\}$.
- $\mu(w) \in M \cup \{w\}$.
- $\mu^2 = Id$.

We refer to $\mu(x)$ *as the mate of* x.

Notice that a matching is a map of order two, that means that if a man m *is matched to a woman* w *then the woman* w *is matched to man* m.

Definition 1.1.4. *We say that an agent* x *is acceptable for an agent* y *if* $x >_y y$.

The first requirement for a good matching will be that the members of each pair are mutually acceptable.

Definition 1.1.5. *The matching* μ *is individually rational if each agent is acceptable to his or her mate. That is, a matching is individually rational if it is not blocked by any agent.*

Note that, no matter what preferences the agents have, at least one individually rational matching will exist, since the matching that leaves every agent single is always individually rational. But these types of matchings are not likely to tell us much. The following definition gives us the matching that will persist over the time and will tell us much more than that where all agents remain singles.

Definition 1.1.6. *A matching* μ *is blocked by a pair* (m, w) *if* $m >_w \mu(w)$ *and* $w >_m \mu(m)$.

Definition 1.1.7. *A matching* μ *is stable if it is not blocked by any individual or any pair of agents, that is:*

- μ *is individually rational.*
- $\forall m \in M$ *and* $w \in W$ *such that* $\mu(m) \neq w$,
  - $\mu(m) \geq_m w$ *or,*
  - $\mu(w) \geq_w m$.

On the contrary, we say a matching μ *is blocked by a pair* (m, w) *such that* $\mu(m) \neq w$ *if* $m >_w \mu(w)$ *and* $w >_m \mu(m)$.

Example. *Consider a marriage market where there are four men and four women, with the list of preferences in 1.1*

All possible matchings are individually rational, since all pairs (m, w) *are mutually acceptable. We affirm that the following matching is stable:*

$$\mu = ((m_1, w_4), (m_2, w_3), (m_3, w_2), (m_4, w_1))$$

Stability may be verified by considering each man in turn as a potential member of a blocking

$P(m_1)$	w_2	w_4	w_1	w_3	$P(w_1)$	m_2	m_1	m_4	m_3
$P(m_2)$	w_3	w_1	w_4	w_2	$P(w_2)$	m_4	m_3	m_1	m_2
$P(m_3)$	w_2	w_3	w_1	w_4	$P(w_3)$	m_1	m_4	m_3	m_2
$P(m_4)$	w_4	w_1	w_3	w_2	$P(w_4)$	m_2	m_1	m_4	m_3
Men's preferences					Women's preferences				

Figure 1.1: Preference list on a marriage market

pair. We can observe that m_1 could form a blocking pair only with w_2 but she prefers her partner so, this is not a blocking pair. Men m_2 and m_3 will never form part of a blocking pair since each of them is matched to his most preferred woman. Finally m_4 only could block together with w_4 but w_4 prefers her partner rather than m_4. So the matching μ is stable.

How to find stable matchings given a marriage market, and the consequences of this outcome, will be studied in the following sections.

1.1.2 Deferred acceptance algorithm

In this section we are going to discuss an algorithm, which produces a stable matching starting from any preference list. As we have seen in the previous example, it could be lengthy to prove if a given matching μ is stable or not, particularly for large markets. We are going to described the original version given by Shapley [1] and Gale[2].

We recall that in the previous section we have ensured the existence of, at least, one matching, the one with all agents remaining singles but this may be unstable.

Theorem 1.1.1. *(Gale and Shapley) A stable matching exists for every marriage market.*

Proof. We shall proof existence by giving an iterative procedure for actually finding a stable set of marriages.

To start, let each man propose to his favourite woman, which is the first woman on his preference list of acceptable women. Each woman rejects the proposal of any man who is not acceptable to her, and each woman who receives more than one proposal rejects all but her most preferred of these. Any man whose proposal is not rejected at this point is kept engaged. However he could be rejected on the next stage if this woman receives a better proposal.

[1] Lloyd Stowell Shapley (June 2, 1923) is a distinguished American mathematician and Nobel Prize winning economist. He is a Professor Emeritus at University of California, Los Angeles (UCLA), affiliated with departments of Mathematics and Economics. He has contributed to the fields of mathematical economics and especially game theory.

[2] David Gale (December 13, 1921 - March 7, 2008) was a distinguished American mathematician and economist. He was a professor emeritus at the University of California, Berkeley, affiliated with the departments of Mathematics, Economics, and Operations Research. He has contributed to the fields of mathematical economics, game theory, and convex analysis.

At any step any man who was rejected at the previous step proposes to his next choice, as long as there remains an acceptable woman to whom he has not yet proposed. Each woman receiving proposals rejects any from unacceptable men, and also rejects all but her most preferred among the group consisting of the new proposers together with any man she may have kept engaged from the previous step.

The algorithm stops after any step in which no man is rejected. We can ensure that algorithm ends since the set of women is finite and in the procedure a man never proposes to a woman that previously rejected him.

At the end of the algorithm every man is either engaged to some woman or has been rejected by every woman on his list of acceptable woman, so he is single. In the same way, women who did not receive any acceptable proposal remain single too.

This procedure ensures that the outcome matching is individually rational, since no man or woman is ever engaged to an unacceptable partner.

Moreover the matching that results is not blocked by a pair. Indeed assume (m, w) blocks μ, then m prefers w to his partner $\mu(m)$, $w >_m \mu(m)$ and w prefers m to $\mu(m)$, $m >_w \mu(w)$. But then, m has proposed w before proposing $\mu(m)$ and has been rejected. If at that moment w was not engaged, she would have accepted and not replaced him by $\mu(w)$. If w was engaged at that moment and her partner was not preferred to $\mu(w)$, he would also been not preferred to m. Hence w would have accepted m when he proposed, which is a contradiction. $\square$

We call this algorithm a "deferred acceptance" procedure, to emphasize the fact that women are able to keep the best available man at any step engaged, without accepting him outright.

The outcome of the deferred acceptance algorithm (DAA) with men proposing is denoted by μ_M and we write μ_W when women propose. This two stable matchings will not typically be the same.

At this point we have to introduce some more notation that allow us to compare between different matchings.

Let μ and μ' be two stable matchings. We denote $\mu >_M \mu'$ if:

- $\mu(m) \geq \mu'(m)$ for all $m \in M$ and,
- $\mu(m) >_M \mu'$ for at least one man $m \in M$.

In a precisely similar way, we define $\geq_W$ or $>_W$ to represent the common preferences of the women over alternative matchings.

Definition 1.1.8. *For a given marriage market (M, W, P), a stable matching μ is M-optimal if every man likes it at least as well as any other stable matching, that is for every other stable matching μ', $\mu \geq_M \mu'$. Analogously we can define W-optimal matches.*

Actually when all agents have strict preferences, there are systematic elements of common interests among the men (and among the women), even in cases in which all men are competing for the same woman, and all women are competing for the same man.

Definition 1.1.9. *An agent x is achievable for an agent y if $\mu(x) = y$ for some μ stable.*

Definition 1.1.10. *We say that agents have strict preferences if nobody is indifferent between several options.*

Theorem 1.1.2. *(Gale and Shapley) When all men and women have strict preferences there always exists an M-optimal stable matching, and a W-optimal stable matching. Furthermore, the matching μ_M produced by the deferred acceptance algorithm with men proposing is the M-optimal stable matching. The W-optimal stable matching is the matching μ_W produced by the algorithm when women propose.*

Proof. We are going to show that, when all men and women have strict preferences, in the deferred acceptance algorithm no man is ever rejected by an achievable woman. Consequently, the stable matching μ_M matches each man to his most preferred achievable woman and, hence, μ_M is M-optimal. We are going to prove it by induction:

- *Step* 1*:* It is clear that in the first step no man is rejected by any achievable woman.
- *Step k:* Assume that in the $k-1$th step in the procedure no man has yet been rejected by a woman who is achievable for him. At this step, suppose woman w rejects man m. If she rejects m as unacceptable, then she is unachievable for him, and we are done. If he is acceptable to her but she rejects him in favour of m', whom she keeps engaged, then she prefers m' to m, $m' >_w m$. We must show that w is not achievable for m.

  We know that m' prefers w to any woman except for those who have previously rejected him, and hence (by the induction assumption) are unachievable for him. Consider an hypothetical matching μ that matches m to w and everyone else to an achievable mate, we have $m' >_w m = \mu(w)$. Then m' prefers w to his mate at μ, $w >'_m \mu(m')$, since preferences are strict, so the matching is unstable because it is blocked by the pair (m', w).

  Therefore there is no stable matching that matches m and w, and so they are unachievable for each other. Hence, $\mu_M(m)$ is the most preferred among all achievable woman and we finally conclude that the matching μ_M is M-optimal.

The proof is analogous when women propose. □

In the previous theorem we proved that the agents on one side of the market have a common interest regarding the set of stable matchings, since they are in agreement on the best stable matching. But now, we are going to prove that agents on the opposite site of the market have opposite interests in this regard, this is that the optimal stable matching for one side of the market is the worst stable matching for the other site of the market.

Theorem 1.1.3. *(Knuth [3]) When all agents have strict preferences, the common preferences of the two sides of the market are opposed on the set of stable matchings: if μ and μ' are stable matching, then all men like μ at least as well as μ' if and only if all women like μ' at least as well as μ. That is, $\mu >_M \mu'$ if and only if $\mu' >_W \mu$.*

Proof. We shall prove both implications:

$\Rightarrow$) Let μ and μ' be stable matchings such that $\mu >_M \mu'$. We will show that $\mu' >_W \mu$.

Suppose that it is not true, so $\mu \geq_W \mu'$. Then, there must be at least one woman w who strictly prefers μ to μ'. Obviously this woman has a different mate at μ and μ' and due to the fact that all stable matchings are individually rational, if w prefers $\mu(w)$ to $\mu'(w)$ then w is not single at μ. consequently $m = \mu(w)$.

Man m, who has also strict preferences, and woman w form a blocking pair for the matching μ': $m = \mu(w) >_w \mu'(w)$ and $w = \mu(m) >_m \mu'(m)$, and this contradicts the assumption that μ' is stable, so $\mu' >_W \mu$.

$\Leftarrow$) Due to the symmetry of the marriage market the proof is analogous.

□

Corollary 1.1.4. *When all agents have strict preferences, the M-optimal stable matching is the worst stable matching for the women and the W-optimal stable matching matches each man with his least preferred achievable mate, this is, it is the worst for the men.*

Example. *Consider a marriage market where there are five men and four women, with the preference list in 1.2. Applying the deferred acceptance algorithm we can find, as we demonstrate, a M-optimal and W-optimal stable matchings.*

$P(m_1)$	w_1	w_2	w_3	w_4	$P(w_1)$	m_2	m_3	m_1	m_4	m_5
$P(m_2)$	w_4	w_2	w_3	w_1	$P(w_2)$	m_3	m_1	m_2	m_4	m_5
$P(m_3)$	w_4	w_3	w_1	w_2	$P(w_3)$	m_5	m_4	m_1	m_2	m_3
$P(m_4)$	w_1	w_4	w_3	w_2	$P(w_4)$	m_1	m_4	m_5	m_2	m_3
$P(m_5)$	w_1	w_2	w_4							
Men's preferences					Women's preferences					

Figure 1.2: Preference list on a marriage market

$$\mu_M = ((m_1, w_1), (m_2, w_2), (m_3, w_3), (m_4, w_4), (m_5))$$
$$\mu_W = (m_1, w_4), (m_2, w_1), (m_3, w_2), (m_4, w_3), (m_5))$$

[3] Donald Ervin Knuth (January 10, 1938) is an American computer scientist, mathematician, and professor emeritus at Stanford University. He is the author of the multi-volume work The Art of Computer Programming. Knuth has been called the "father of the analysis of algorithms".

In this example we have verified Theorem 1.1.2, all men like μ_M at least as well as μ_W and all women like μ_W at least as well as μ_M. Besides, it is obvious that the matchings are not the same.

In appendix A, we propose a code in C to compute the matching resulting of applying the Deferred Acceptance algorithm when men are proposing. It has been necessary to add some assumptions in order to simplify the problem.

1.2 The lattice property

In what follows we are going to assume that all agents have strict preferences. In the previous section we saw that there always exist at least one stable matching and, if this matching is produced by the deferred acceptance algorithm with men proposing is the M-optimal stable matching, and if are women who propose it is a W-optimal stable matching. We finished the section mentioning that the M-optimal matching is the worst matching for women. Due to this properties we are going to be able to produce new stable matchings from two given stable matchings and we shall prove that the set of the stable matchings forms a distributive lattice under a natural ordering relation, and the M-optimal and W-optimal matches are the infimum and the supremum of the lattice.

Let μ and μ' be two stable matchings such that, $\mu,\ \mu' :\quad M \cup W \longrightarrow M \cup W$. We can define:

$$\lambda := \mu \vee_M \mu' = \begin{cases} \lambda(m) = \mu(m) & if \quad \mu(m) >_m \mu'(m) \\ \lambda(m) = \mu'(m) & if \quad \mu(m) \leq_m \mu'(m) \end{cases}$$

We can observe that this function assigns each man his more preferred mate from μ and μ', and consequently it assigns each woman her less preferred mate. Define now,

$$\upsilon := \mu \wedge_M \mu' = \begin{cases} \upsilon(m) = \mu(m) & if \quad \mu(m) <_m \mu'(m) \\ \upsilon(m) = \mu'(m) & if \quad \mu(m) \geq_m \mu'(m) \end{cases}$$

This function assigns each man his less preferred mate and it assigns each woman her more preferred mate.

Theorem 1.2.1. ***The Lattice theorem.*** *(Conway[4]) When all preferences are strict, if μ and μ' are stable matchings, then the functions $\lambda = \mu \vee_M \mu'$ and $\upsilon = \mu \wedge_M \mu'$ are both matchings. Furthermore, they are both stable.*

Proof. First, we are going to prove that both functions are matchings, in fact we prove it only for the λ function, for the υ function the argument is symmetric. After that we prove that both matchings are stable.

[4]John Horton Conway (December 26, 1937) is a British mathematician active in the theory of finite groups, knot theory, number theory, combinatorial game theory and coding theory. Conway is currently Professor of Mathematics and John Von Neumann Professor in Applied and Computational Mathematics at Princeton University

- *Matchings:* Suppose that man m and m' received the same partner w according to λ, this is possible if (m, w) is a pair in μ and (m', w) is a pair in μ'. Then $\mu >_m \mu'$ and $\mu' >_{m'} \mu$. Then Theorem 1.1.3 applied to the pair (m, w) implies that $m' >_w m$ and applied to the pair (m', w) implies that $m >_w m'$ giving a contradiction.

  Hence, a matching does result.

- *Stability:* Suppose that λ is not stable, then it exists at least one pair (m, w) that blocks the matching. This is, $w >_m \lambda(m)$ from which it follows, from the definition of λ that $w >_m \mu(m)$ and $w >_m \mu'(m)$. On the other hand $m >_w \lambda(w)$. Hence, if $\lambda(w) = \mu(w)$ then (m, w) blocks μ, and if $\lambda(w) = \mu'(w)$ then (m, w) blocks μ'.

  In either case we get a contradiction, since μ and μ' are both stable.

□

The existence of M-optimal and W-optimal stable matchings can be deduced from this theorem and due to the fact that there are only a finite number of stable matchings.

Definition 1.2.1. *A lattice is a partially ordered set L in which every two elements a, b have a supremum, denoted by $a \vee b$ and an infimumm denoted by $a \wedge b$. A lattice L is complete when each of its subsets X has a supremum and an infimum in L.*

We have introduced the notation of lattice to testify that the lattice theorem, as its name indicates, demonstrates that the set of stable matchings is a lattice. This observation is important because it gives to our set of stable matchings an algebraic structure.

Theorem 1.2.2. *When preferences are strict, the set of stable matchings is a distributive lattice under the common order of the men, dual to the common order of the women.*

Once we have seen that the set of stable matchings is a distributive lattice, we can ask ourselves if it is a distributive lattice with particular property.

Theorem 1.2.3. *Every finite distributive lattice equals the set of stable matchings of some marriage market.*

Since the lattice of stable matchings is not a particular one but any distributive lattice can be shown to be a lattice of stable matchings this line of investigation will not bear any further fruit.

In section 1.5 we are going to give an algorithm to compute all possible stable matchings and we are going to illustrate it by an example using the theory developed in this section.

1.3 Weak Pareto efficiency

In this new section we want to develop new theory focused on marriage markets with strict preferences. We are going to prove again almost all our previous obtained results, using only properties about stability and optimality.

In what follows optimality an efficiency will refer to the same concept and would be used interchangeably.

We will need some notation to discuss what happens when a man or a woman extends his (her) list of acceptable women (men) by adding people to the end of the original list of acceptable partners.

We will write $P'_m \geq P_m$ if P'_m is such an extension of P_m, and we will write $P' \geq_M P$ if $P'_m \geq P_m \; \forall m \in M$. Similarly for women.

Theorem 1.3.1. ***Decomposition Lemma*** *(Gale and Sotomayor). Let μ and μ' be, respectively, two stable matchings in markets (M, W, P) and (M, W, P') with $P' \geq_M P$, and all preferences are strict. Let $M(\mu')$ be the set of men who prefer μ' to μ under P and let $W(\mu)$ be the set of women who prefer μ to μ' under P. Then μ' and μ map $M(\mu')$ onto $W(\mu)$. That is, μ' and μ match any man who prefers μ' to a woman who prefers μ, and vice versa.*

Proof. Suppose $m \in M(\mu')$, then, by definition of the set $M(\mu')$, $\mu'(m) >_m \mu(m) \geq_m m$, under P so $\mu'(m) \in W$. Setting $w = \mu'(m)$, we cannot have $\mu'(w) >_w \mu(w)$ for then (m, w) would block μ. Hence, since preferences are strict, $w \in W(\mu)$ and so $\mu'(M(\mu'))$ is contained in $W(\mu)$.

On the other hand, if $w \in W(\mu)$ then $\mu(w) >_w \mu'(w) \geq_w w$, so $\mu(w) \in M$. Letting $\mu(w) = m$, we see that we cannot have $\mu(m) >_m \mu'(m)$ under P' because (m, w) would block μ'. Hence, since the preferences are strict, $\mu'(m) >_m \mu(m) = w >_m m$ under P' and P so $m \in M(\mu')$ and $\mu(W(\mu))$ is contained in $M(\mu')$.

Since μ and μ' are one-to-one and $M(\mu)$ and $W(\mu')$ are finite, the conclusion follows. □

Corollary 1.3.2. ***Decomposition lemma when P = P'.*** *Let μ and μ' be stable matchings in (M, W, P), where all preferences are strict. Let $M(\mu)$ be the set of men who prefer μ to μ' and $W(\mu)$ the set of women who prefer μ to μ'. Analogously define $M(\mu')$ and $W(\mu')$. Then $\mu : M(\mu') \mapsto W(\mu)$ and $\mu' : M(\mu) \mapsto W(\mu')$.*

We can observe that this result allows us to give an alternate proof of the Theorem 1.1.3.

Proof. Alternative proof of Theorem 1.1.3: $\mu' >_M \mu$ under P if and only if $M(\mu)$ is empty and $M(\mu')$ is non-empty. This is equivalent to $\mu(m) = \mu'(m)$ for all $m \in M - M(\mu')$ and $W(\mu)$ is non-empty, which in turn is satisfied if and only if $\mu(w) = \mu'(w)$ for all $w \in W - W(\mu)$ and $\mu(w) >_w \mu'(w)$ for some w, and this is equivalent to $\mu >_W \mu'$. □

In the next theorem we are concerned about the particular group of people who remain single. Its proof is an immediate consequence of the decomposition lemma when $P = P'$.

Theorem 1.3.3. *In a market (M, W, P) with strict preferences, the set of people who are single is the same for all stable matchings.*

Proof. Suppose m was matched under μ', so m is not single, and unmatched under μ, so m is single. Then, $m \in M(\mu')$, but from the decomposition lemma with $P = P'$, μ maps $W(\mu)$ onto $M(\mu')$, so m is also matched under μ, which is a contradiction. □

Similarly, the lattice property of the set of stable matchings can also been proved by means of the decomposition lemma. For a detailed proof see [11].

The next theorem takes another look at the sense in which the M-optimal stable matching is optimal for all men. We have already studied the sense in which it is as good a stable matching as the men can achieve, but now we want to ask whether there might not be some other unstable matching that all the men would prefer.

Theorem 1.3.4. ***Weak Pareto*** [5] ***optimality for the men.*** *There is no individually rational matching μ (stable or not) such that $\mu >_m \mu_M$ for all $m \in M$*

Proof. We shall prove it using the deferred acceptance algorithm with some additional assumptions:

1. We are going to suppose that $\#M = \#W$, that is, there are as many men as women in the market.
2. All possible matchings are going to be individually rational, that is, the preference profile is such that all pairs (m, w) are mutually acceptable.

With this limitation the deferred acceptance algorithm (for men proposing) has an important property that it has not in the general case:

If w is the last woman to become engaged, no man is rejected by w (since the algorithm terminates when the last woman receives her first proposal).

Now, we can start the proof.

Clearly there can be no stable matching μ with the property stated, since μ_M is optimal among all stable matchings. Suppose that there exists an unstable matching μ such that $\mu >_M \mu_M$. If w is the last woman to become engaged by the deferred acceptance algorithm, due to the property mentioned, no man has been rejected by w.

Let us suppose that $\mu_M(w) = m$ and $m' = \mu(w)$, then, by the assumption $\mu(m') = w >_{m'} \mu_M(m')$, so w must have rejected m' which is a contradiction. □

We have seen that the optimal stable matching for one side of the market is weakly Pareto optimal for the agents on that side of the market. This means there is no matching that all those agents prefer.

We are going to give an example to show that it can exist a matching μ, unstable, such that: $\mu(m) \geq_m \mu_M(m)\ \ \forall m \in M$ and $\exists m'$ such that $\mu >_{m'} \mu_M$.

Example. *Consider the example of a particular marriage market with preferences detailed in 1.3 .*

Then, using the deferred acceptance algorithm:

$$\mu_M = ((m_1, w_1), (m_2, w_3), (m_3, w_2))\,.$$

Nevertheless,

$$\mu = ((m_1, w_2), (m_2, w_3), (m_3, w_1))\,.$$

[5] Vilfredo Federico Damaso Pareto (15 July 1848 - 19 August 1923) was an Italian economist who first focused on the kind of optimality where a collective prefers optimality to stability

$P(m_1)$	w_2	w_1	w_3	$P(w_1)$	m_1	m_2	m_3
$P(m_2)$	w_1	w_2	w_3	$P(w_2)$	m_3	m_1	m_2
$P(m_3)$	w_1	w_2	w_3	$P(w_3)$	m_1	m_2	m_3
Men's preferences				Women's preferences			

Figure 1.3: Preference list on a marriage market

leaves m_2 no worse than under μ_M, but benefits m_1 and m_3. We, obviously, can observe that μ is not stable since,

$$w_1 >_{m_2} \mu(m_2) = w_3$$

$$m_2 >_{w_1} \mu(w_1) = m_3$$

and, (m_2, w_1) blocks μ.

1.4 The core

We have already analysed the structure of the set of stable matchings, to see it has a lattice structure, which has lead to several interesting consequences regarding optimal stable matchings and opposition of interests between the two sides of the market.

Now, we analyse whether stable matchings can be blocked by bigger coalitions. This is related to the notion of core, which is a basic notion in other coalitional problems.

In a more general setting of coalitions games, the rules of the game together with the specific preferences of the players induce a relation on the outcomes, called domination relation.

Definition 1.4.1. *For any two feasible outcomes x and y, x dominates y if and only if there exists a coalition of players S such that:*

- *Every member of the coalition S prefers x to y.*
- *The rules of the game give the coalition S the power to enforce x (over y).*

This is, x dominates y if there is some coalition S whose members have both the incentive and the means to replace y with x. For this reason we might expect that y will not be the outcome of the game.

Given a dominance relation among outcomes, we define the core of the game. Notice that the notion of core can be considered in many different settings, whenever a dominance relation among outcomes is defined.

Definition 1.4.2. *The core of a game is the set of undominated outcomes.*

It is important to note the difference between the definition of the core and the definition of the set of the stable matchings. The first one, the core, is defined via a domination relation in which all coalitions play a potential role, whereas the second one, the set of stable matchings, is defined with respect to certain kinds of coalitions, individual and mixed-pairs only.

That is, an outcome fails to be in the core if it is blocked by any coalition of agents, whereas it fails to be a stable matching only if it is blocked by some individual agent or by some pair of agents.

Let us now introduce a dominance relation in the setting of matchings in a market.

Definition 1.4.3. *A matching μ' dominates another matching μ if and only if there exists a coalition $A \in W \cup M$, such that, for all $m, w \in A$:*

- $\mu'(m) \in A$ *and* $\mu'(w) \in A$.
- $\mu'(m) >_m \mu(m)$ *and* $\mu'(w) >_w \mu(w)$.

The following theorem shows that for the marriage market, nothing has been lost by ignoring coalitions other than singletons and pairs.

Theorem 1.4.1. *The core of the marriage market equals the set of stable matchings.*

Proof. We shall prove both directions:

$\subset$ We shall prove that any matching in the core is stable. Let μ be in the core and let us see it is stable. If μ is individually irrational, then it is dominated via a singleton coalition, and if its unstable via some man m and woman w, with $m >_w \mu(w)$ and $w >_m \mu(m)$, then it is dominated via the coalition $\{m, w\}$ by any matching μ' with $\mu'(m) = w$.

$\supset$ We shall proof that stable matchings are in the core. Take a stable matching μ. If μ is not in the core, then μ is dominated by some matching μ' via a coalition A. This is,

- $\forall m \in A$, $\mu'(m) >_m \mu(m)$.
- $\forall w \in A$, $\mu'(m) >_w \mu(m)$.
- If $m \in A$ and $w \in A$ then $\mu'(w) \in A$ and $\mu'(m) \in A$.

If μ is individually irrational then μ is not stable and we reach a contradiction.

Assume μ is individually rational, then: $\forall m \in A$, $\mu'(m) >_m \mu(m) \geq_m m$ so $\mu'(m) >_m m$ and $\forall w \in A$, $\mu'(w) >_w \mu(w) \geq_w w$ so $\mu'(w) >_w w$. We can conclude that $\mu'(m) \in W$ and $\mu'(w) \in M$.

Take any $w \in A$, as we just have proved, $\mu'(w) = m \in A$. Let us see that (m, w) blocks μ.

- $\mu'(m) >_m \mu(m)$ because m is in the coalition A.
- $\mu'(w) >_w \mu(w)$ because w is in the coalition A.

So, such μ is blocked, by the pair (m, w) and μ is no stable which is a contradiction.

$\square$

The above theorem shows that if a matching μ is not stable, then it is not in the core of the market; that is, there is some matching μ' that dominates μ.

We will see in the next chapter that for two-sided markets where matchings are not one-to-one but many-to-one, the set of stable matchings and the core may differ.

1.5 Computational questions

In this section we turn to some computational questions about the marriage model. We are going to focus on marriages where agents have strict preferences. We are going to return to the lattice property of a market and we will be able to give an algorithm to compute every stable matching given a market. Moreover, we are going to study the number of stable matchings and find a lower and an upper bound for this number.

1.5.1 An algorithm to compute every stable matching

We shall consider that P is a preference profile then $P(x)$ will be the list of acceptable people of the agent x, with the inclusion of x as the last entrance.

From the optimality of μ_M and μ_W follows that if (m, w) is a mutually acceptable pair and $w >_m \mu_M(m)$, or $m >_w \mu_W(w)$, then m and w cannot be matched at any stable matching. This suggest that the preference lists can be shortened using a reduction procedure and without changing the set of stable matchings. For all $m \in M$ and $w \in W$:

Step 1: Remove from m's list of acceptable women all $w \in W$ who are more preferred than $\mu_M(m)$. Do the same for each woman.

Thus, $\mu_M(m)$ will be the first entry in m's reduced list and $\mu_W(w)$ will be the first entry in $w's$ reduced list.

Step 2: Remove from w's list of acceptable men all who are less preferred than $\mu_M(w)$ (we here apply Corollary 1.1.4). Remove from m's list of acceptable women all who are less preferred than $\mu_W(m)$.

Thus, $\mu_M(w)$ will be the last entry in w's reduced list and $\mu_W(m)$ will be the last entry in m's reduced list.

Step 3: After steps one and two, if m is not acceptable to w (i.e., if m is not on w's preference list as now modified), then remove w from $m's$ list of acceptable women, and similarly remove from w's list of acceptable men any man m to whom w is no longer acceptable.

We have, finally, that m will be acceptable to w if and only if w is acceptable to m after step three.

In general, if μ is any stable matching and we replace μ_M by μ in the reduction process described, the resulting profile will be called a profile of reduced lists for the original market and will be denoted by $P(\mu)$.

Properties: We can deduce from the construction of $P(\mu)$:

- $\mu(m)$ is the first entry of $P(\mu)(m)$ and $\mu(w)$ is the last entry of $P(\mu)(w)$. $\mu_W(m)$ is the last entry of $P(\mu)(m)$ and $\mu_W(w)$ is the first entry of $P(\mu)(w)$.
- μ is the M-optimal stable matching under $P(\mu)$ and μ_W is the W-optimal stable matching under $P(\mu)$.
- m is acceptable to w if and only if w is acceptable to m under $P(\mu)$.
- If a man or woman is the only acceptable man or woman for someone on the other side of the market, then he or she is not acceptable to anyone else.

Definition 1.5.1. *A set of men $\{a_1, \cdots, a_r\}$ defines a cycle for some profile of reduced lists, $P(\mu)$ if:*

- *For $i = 1, \cdots, r-1$, the second woman in $P(\mu)(a_i)$ is $\mu(a_{i+1})$, that is the first woman in $P(\mu)(a_{i+1})$.*
- *The second woman in $P(\mu)(a_r)$ is $\mu(a_1)$, that is the first woman in $P(\mu)(a_1)$.*

We denote such a cycle by $\sigma = (a_1, \cdots, a_r)$ and we say that a_i generates the cycle σ for any $i = 1, \cdots, r$.

Example. *Consider the preference profile given in 1.4:*

$P(\mu)(a_1)$	w_3	w_5	w_2
$P(\mu)(a_2)$	w_5	w_2	$\cdots$
$P(\mu)(a_3)$	w_2	w_3	$\cdots$

Figure 1.4: Preference list on a marriage market

We can see that the second woman in $P(\mu)(a_1)$ is w_5 which is the first woman in $P(\mu)(a_2)$; w_2 is the second woman in $P(\mu)(a_2)$ and is the first woman in $P(\mu)(a_3)$. Finally w_3 is both the first woman in $P(\mu)(a_1)$ and the second woman in $P(\mu)(a_3)$.

Hence $\sigma = (a_1, a_2, a_3)$ is a cycle.

It is clear that we can only have a cycle if $P(\mu)(m)$ has more than one acceptable woman for some m. In the previous example we can find a cycle without using any algorithm, but sometimes, when we have large markets, it is useful to define a procedure to find it.

Let p_1 be an arbitrary man such that $P(\mu)(p_1)$ contains more than one woman. Then we can construct an oriented graph, whose nodes are $M \cup W$, as follows:

- There is an arc from p_i to q_{i+1} if q_{i+1} is the second woman in $P(\mu)(p_i)$.
- There is an arc from q_i to p_i if p_i is the last man in $P(\mu)(q_i)$, that is $\mu(q_i) = p_i$.

This graph will close at some step (when some q_s or p_s is repeated) and then we will get a cycle.

Definition 1.5.2. *Let μ a stable matching and $\sigma = (a_1, \cdots, a_r)$ a cycle for $P(\mu)$. Then* *we can define a matching μ', called cyclic matching under $P(\mu)$, as follows:*

$$\mu'(a_i) = \mu(a_{i+1}) \ \forall i = 1, \cdots, r-1.$$
$$\mu'(a_r) = \mu(a_1).$$
$$\mu'(m) = \mu(m) \ \forall m \notin \sigma.$$

Proposition 1.5.1. *Let $P(\mu)$ be a profile of reduced lists. If μ' is a cyclic matching under $P(\mu)$, then μ' is stable under the original preferences.*

To prove this proposition we need some more theory that we are not going to develop in this monograph because it exceeds the main aim of this section. For this reason we made the statement here without proof.

We now can develop an algorithm to find every stable matching given a marriage market:

> *Step* 1 Find, by the deferred acceptance procedure, μ_M, μ_W and $P(\mu_M)$.
>
> *Step* k For each profile P' of reduced lists obtained in step $(k-1)$, find all corresponding cycles and for each cycle obtain the corresponding cyclic matching for P'. Then for each cyclic matching μ obtain the profile of reduced lists $P(\mu)$.

This algorithm stops after a finite number of steps, as next proposition shows.

Proposition 1.5.2. *The algorithm stops at step n if and only if we obtain only one profile of reduced lists in this step and the men's lists of acceptable women have at most one woman.*

Proof. If the algorithm stops at step t, this means that every profile of reduced lists obtained in this step has no cycles, otherwise n would not be the last step. Then there is at most one acceptable woman in each man's lists since there is a cycle for $P(\mu)$ if and only if $P(\mu)(m)$ has more than one acceptable woman for some m. On the other hand, the M-optimal matching under each one of these profiles must be the W-optimal matching under the original preferences, which shows that all these profiles must be the same.

Hence, we have proved that this algorithm stops after a finite number of steps. □

Example. *In section 1.2 we have proved that the set of stables matchings has a lattice structure. Now, we have enough tools to show how we can find every matching. We shall follow the algorithm described along this section. In 1.5 we have the preference list of the agents of the market.*

$P(m_1)$	w_1	w_2	w_3	w_4	$P(w_1)$	m_4	m_3	m_2	m_1
$P(m_2)$	w_2	w_1	w_4	w_3	$P(w_2)$	m_3	m_4	m_1	m_2
$P(m_3)$	w_3	w_4	w_1	w_2	$P(w_3)$	m_2	m_1	m_4	m_3
$P(m_4)$	w_4	w_3	w_2	w_1	$P(w_4)$	m_1	m_2	m_3	m_4
Men's preferences					Women's preferences				

Figure 1.5: Preference list on a marriage market

First of all we have to find the M-optimal and W-optimal stable matchings to compute the reduced profile list of each agent. We use the deferred acceptance procedure, although it can be seen without using the algorithm.

$$\mu_M = ((m_1, w_1), (m_2, w_2), (m_3, w_3), (m_4, w_4))$$
$$\mu_W = ((m_1, w_4), (m_2, w_3), (m_3, w_2), (m_4, w_1))$$

We can observe that the list of preferences cannot be reduced because the first entrance of each man is $\mu_M(m)$ and the worst is $\mu_W(m)$. Similarly for women.

Now, we want to find a cycle. Consider man m_1. His second choice is w_2 who is the first choice for m_2 and m_2's second choice is w_1 who is the first choice for m_1. This means that $\sigma_1 = (m_1, m_2)$ form a cycle. So we have another stable matching:

$$\mu_1 = ((m_1, w_2), (m_2, w_1), (m_3, w_3), (m_4, w_4))$$

Analogously $\sigma_2 = (m_3, m_4)$ is a cycle.

$$\mu_2 = ((m_1, w_1), (m_2, w_2), (m_3, w_4), (m_4, w_3))$$

With these two new stable matchings, by using the lattice structure, we can generate:

$$\mu_3 = \mu_1 \wedge_M \mu_2 = ((m_1, w_2), (m_2, w_1), (m_3, w_4), (m_4, w_3))$$

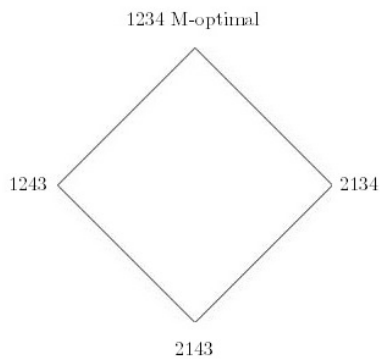

Figure 1.6: First sub-lattice structure

Now, taking as reference matching μ_3, we built the reduced lists $P(\mu_3)$ by deleting in each preference list $P(\mu_i)$ the first woman (and as consequence the last man in woman preference list $P(w_i)$.

We see that the resulting profile, profile 1.7, has two more cycles: $\sigma_4 = (m_1, m_4)$ and $\sigma_5 = (m_2, m_3)$. So we have two more stable matchings:

$$\mu_4 = ((m_1, w_3), (m_2, w_1), (m_3, w_4), (m_4, w_2))$$

$P(m_1)$	w_2	w_3	w_4	$P(w_1)$	m_4	m_3	m_2
$P(m_2)$	w_1	w_4	w_3	$P(w_2)$	m_3	m_4	m_1
$P(m_3)$	w_4	w_1	w_2	$P(w_3)$	m_2	m_1	m_4
$P(m_4)$	w_3	w_2	w_1	$P(w_4)$	m_1	m_2	m_3
Men's preferences				Women's preferences			

Figure 1.7: Profile of reduced list $P(\mu_3)$

$$\mu_5 = ((m_1, w_2), (m_2, w_4), (m_3, w_1), (m_4, w_3))$$

With these two new stable matchings we can generate:

$$\mu_6 = \mu_4 \wedge_M \mu_5 = ((m_1, w_3), (m_2, w_4), (m_3, w_1), (m_4, w_2))$$

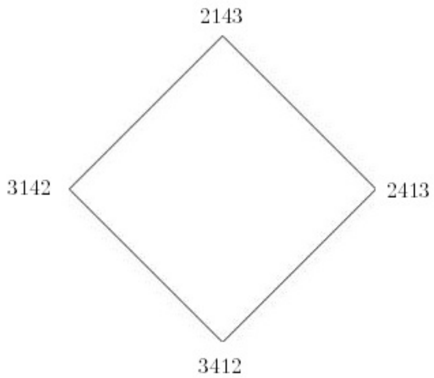

Figure 1.8: Second sub-lattice structure

We now consider the reduced profile $P(\mu_6)$.

$P(m_1)$	w_3	w_4	$P(w_1)$	m_4	m_3
$P(m_2)$	w_4	w_3	$P(w_2)$	m_3	m_4
$P(m_3)$	w_1	w_2	$P(w_3)$	m_2	m_1
$P(m_4)$	w_2	w_1	$P(w_4)$	m_1	m_2
Men's preferences			Women's preferences		

Figure 1.9: Profile of reduced list $P(\mu_6)$

Repeating the procedure with this reduced list, the reduced list shown in 1.9, we find two more cycles, $\sigma_5 = (m_1, m_2)$ and $\sigma_6 = (m_3, m_4)$ that generates, respectively, the following stable matchings:

$$\mu_7 = ((m_1, w_4), (m_2, w_3), (m_3, w_1), (m_4, w_2))$$
$$\mu_8 = (m_1, w_3), (m_2, w_4), (m_3, w_2), (m_4, w_1))$$

With these two matchings we can generate another matching:

$$\mu_9 = \mu_7 \wedge_M \mu_8 = ((m_1, w_4), (m_2, w_3), (m_3, w_2), (m_4, w_1)) = \mu_W$$

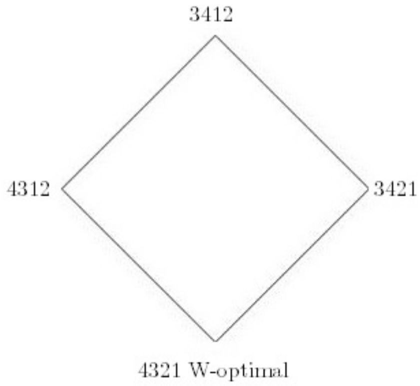

Figure 1.10: Third sub-lattice structure

As we expected the last stable matching generated doing this procedure is the W-optimal stable matching, so we can ensure that we have found all possibles stable matchings for the marriage market given.

1.5.2 The number of stable matchings

At this point, we have seen how to find all stable matchings given a particular market. But this is not always feasible due to the fact that the number of stable matchings may grow exponentially with the size of the market. Due to the relation between the size of the market, suppose n, and the number of stable matchings, we will denote the number of stable matchings by $f(n)$.

In this section we are going to consider a given instance of size n, (M, W, P) and we will give a lower bound for $f(n)$. As we remark in previous sections, a lower bound, given any instance of size n is 1. This is, at least one stable matching exists, the one given by the deferred acceptance algorithm.

In this section we are going to add the assumption that the preference list of every agent in the market contains all agents of the opposite sex, that is, remaining single is the worst possibility. As in previous sections, we assume preferences are strict.

A trivial upper bound given an instance of size n is $n!$ but to find a better bound is an open problem.

The following lemma establishes the exponential growth rate of the function $f(n)$.

Lemma 1.5.1. *Given stable marriage instances of sizes m and n with x and y stable matchings respectively, there is an instance of size mn with at least $max(xy^m, yx^n)$ stable matchings.*

Proof. Suppose that men and women in the given instances are labelled as:

$$a_1, \cdots, a_m, c_1, \cdots, c_m \text{ and } b_1, \cdots, b_n, d_1, \cdots, d_n$$

where $a_i,\ b_j \in M \quad \forall i = 1, \cdots, m$ and $\forall j = 1, \cdots n$ and $c_i,\ d_j \in W \quad \forall i = 1, \cdots, m$ and $\forall j = 1, \cdots n$. Consider the instance of size mn in which:

- The men are labelled (a_i, b_j), $i = 1, \cdots, m$, $j = 1, \cdots, n$.
- The women are labelled (c_i, d_j), $i = 1, \cdots, m$, $j = 1, \cdots, n$.
- Man (a_i, b_j) prefers (c_k, d_l) to $(c_{k'}, d_{l'})$ if b_j prefers d_l to $d_{l'}$, or if $l = l'$ then a_i prefers c_k to $c_{k'}$.
- Woman (c_i, d_j) prefers (a_k, b_l) to $(a_{k'}, b_{l'})$ if d_j prefers b_l to $b_{l'}$, or if $l = l'$ then c_i prefers a_k to $a_{k'}$.

Let $\mu_1, \cdots, \mu_n$ be any sequence of (not necessarily distinct) stable matchings in the instance of size m, and let μ be any stable matching in the instance of size n. The total number of choices available for $\mu_1, \cdots, \mu_n$ and μ is clearly yx^n. Then we claim that the mapping

$$(a_i, b_j) \longleftrightarrow (\mu_j(a_i), \mu(b_j))$$

is a stable matching in the composite instance.

It is immediately clear that this mapping is actually a matching, since both μ_j and μ are matchings. So suppose that this matching is blocked by the pair $((a, b), (c, d))$. Then, of the following conditions, and taking into account preferences are assumed to be strict, we must have either (i) or (ii) together with either (iii) or (iv).

i b prefers d to $\mu(b)$.

ii $d = \mu(b)$ and a prefers c to $\mu_j(a)$.

iii d prefers b to $\mu(d)$.

iv $b = \mu(d)$ and c prefers a to $\mu_j(c)$.

Of the four possibilities, the combination of (i) with (iii) is precluded by the stability of μ, (ii) with (iv) by the stability of μ_j, and the others by simple incompatibility.

Hence our claim is justified, and we have demonstrated an instance with at least yx^n stable matchings. Likewise, by interchanging the roles of the original two instances, we establish the corresponding result for xy^m. □

Theorem 1.5.1. *For each $n \geq 1$, n power of 2, there is a stable marriage instance of size n with at least 2^{n-1} stable matchings.*

Proof. We shall prove it by induction:

- *Initial case:* For $n = 2^0$, the trivial instance of size 1, admits a single stable matching. For $n = 2^1$ there exist at least one stable matching, the one obtained applying deferred acceptance algorithm when men propose.
- *Assume it true for $n = 2^k$:* We apply repeatedly the construction of Lemma 1.5.1 with $m = 2$ and the instance of size 2 shown in figure 1.11 below. For this instance, both possible matchings are stable, so that $x = 2$ and, by the inductive hypothesis, $y = 2^{2^k-1}$. Hence, by Lemma 1.5.1, there exists an instance of size $2 \cdot 2^k = 2^{k+1}$ with at least $max(2 \cdot (2^{2^k-1})^2, 2^{2^k-1} \cdot 2^{2^k}) = 2^{2^{k+1}-1}$ stable matchings, as required.

$P(m_1)$	w_1	w_2	$P(w_1)$	m_2	m_1
$P(m_2)$	w_2	w_1	$P(w_2)$	m_1	m_2
Men's preferences			Women's preferences		

Figure 1.11: Example of an instance of size 2

□

It turns out that Theorem 1.5.1 gives a rather low estimate of the number of stable matchings. For example, we have already seen that the instance of size 4 shown in figure 1.3, has 10 stable matchings rather than 8.

In fact, if n is power of 2, it can be shown (further information in [5]) that if x_n represents the number of stable matchings in the instance of size n so generated, then:

$$x_n = 3x_{n/2}^2 - 2x_{n/4}^4$$

for $n \geq 4$. This recurrence relation leads to the table of values shown in figure 1.12, where the values are compared to the bound 2^{n-1} given by the theorem.

k	$n = 2^k$	2^{n-1}	x_n
1	2	2	2
2	4	8	10
3	8	128	268
4	16	32768	195472

Figure 1.12: Lower bounds for the maximum number of stable matchings

1.6 A remark on other markets

In this section we want to focus on two particular extensions of the stable marriage problem. As we have seen, this problem involves two disjoint sets. Our next cases of study are concerned on markets where we have just one set of participants or when we have three sets to match. The first one is called the room-mate problem and the second one is called the man-woman-child problem.

One of the most significant results we achieve during this work is that, on the marriage stable problem, there always exist a stable matching. Now, we are going to show in each of these two new cases that the preferences of the agents involved may be such that no stable matching exist.

1.6.1 One-sided markets

The room-mate problem is essentially a version of the marriage problem involving just one set, that is a one-sided market. Each person in the set, of even cardinality n, ranks the $n-1$ others in a list of preferences. The object is to find a stable matching, which is a partition of the set into $n/2$ pairs of room-mates such that no two person who are not room-mates prefer each other to their actual partners.

We shall illustrate that not always a stable matching exists giving a simple example described in the paper of Gale and Shapley [4].

Example. *Consider an instance of size* 4 *with the preferences given in 1.13.*

$P(a)$	b	c	d
$P(b)$	c	a	d
$P(c)$	a	b	d
$P(d)$	arbitrary		

Figure 1.13: Preference lists on a one-sided market

We can observe that person d is the last choice of everyone else. Each of the other people is someone else's first choice. So no matching will be stable, since any matching must pair someone with agent d, and that someone will be able to find another person to make a blocking pair. Possible matchings are:

$$\mu_1 = ((c, b), (a, d)) \qquad \mu_2 = ((a, b), (c, d)) \qquad \mu_3 = ((d, b), (a, c)).$$

In each of these matchings we have a blocking pair which are (c, a) *in* μ_1, (c, b) *in* μ_2 *and* (a, b) *in* μ_3.

We can finally conclude using this example that we cannot always find a stable matching on one-sided markets.

In spite of this result, Knuth [6], found an efficient algorithm (polynomial-time in the worst case) to generate a solution, if one exist.

[6]Donald Ervin Knuth (January 10, 1938) is an American computer scientist, mathematician, and Professor Emeritus at Stanford University. He is the author of the multi-volume work The Art of Computer Programming. Knuth has been called the "father" of the analysis of algorithms. He contributed to the development of the rigorous analysis of the computational complexity of algorithms and systematized formal mathematical techniques for it.

1.6.2 Three-sided markets

The child-woman-man problem is essentially an extension of the marriage problem, in which there are three sets of people: men, women and children. In this market, a matching is a division of the people into groups of three, containing one man, one woman and one child. Each person has preferences over the set of pairs he or she might possibly be matched with.

A man, a woman and a child (m, w, c) block a matching if m prefers (w, c) to $\mu(m)$; w prefers (m, c) to $\mu(w)$ and c prefers (m, w) to $\mu(c)$. As in one-sided markets, we shall demonstrate the non existence of a stable matching giving a particular example.

Example. *Consider three men, three women and three children with the preference list detailed in 1.14.*

$P(m_1)$	(w_1, c_3)	(w_2, c_3)	(w_1, c_1)	$\cdots$	arbitrary
$P(m_2)$	(w_2, c_3)	(w_2, c_2)	(w_3, c_3)	$\cdots$	arbitrary
$P(m_3)$	(w_3, c_3)	$\cdots$		arbitrary	
$P(w_1)$	(m_1, c_1)	$\cdots$		arbitrary	
$P(w_2)$	(m_2, c_3)	(m_1, c_3)	(m_2, c_2)	$\cdots$	arbitrary
$P(w_3)$	(m_2, c_3)	(m_3, c_3)	$\cdots$	arbitrary	
$P(c_1)$	(m_1, w_1)	$\cdots$		arbitrary	
$P(c_2)$	(m_2, w_2)	$\cdots$		arbitrary	
$P(c_3)$	(m_1, w_3)	(m_2, w_3)	(m_1, w_2)	(m_3, w_3)	arbitrary

Figure 1.14: Preference list on a three-sided market

We can affirm that no stable matching exists.

1. All matchings that give m_1 (respectively m_2 and w_2) a better family than (m_1, w_1, c_1) (respectively (m_2, w_2, c_2)) are unstable. To see this fact, we can note that any matching containing either (m_1, w_1, c_3) or (m_2, w_2, c_3) is blocked by (m_3, w_3, c_3) and any matching containing (m_1, w_2, c_3) is blocked by (m_2, w_3, c_3).

2. Any matching that does not contain (m_1, w_1, c_1) (respectively (m_2, w_2, c_2)) is either blocked by (m_1, w_1, c_1)(respectively (m_2, w_2, c_2))) or is unstable as already shown in item 1 above.

3. Finally, (m_1, w_2, c_3) blocks any matching that contains (m_1, w_1, c_1) and (m_2, w_2, c_2).

So all matchings are unstable.

Chapter 2

The college admission model

In this chapter we are going to focus on those markets where matchings are many to one. Since there are a large variety of examples of these markets: students and colleges, workers and firms, medical interns and hospitals, we are going two refer one side of agents as colleges and the other one as students.

The main idea of this problem is that each student would like to attend a college and has preferences over colleges and the option of remaining unmatched. Each college would like to recruit a maximum number of students determined by its capacity. Colleges have also individual preferences over students and the option to let a sit unfilled but, in addition, they have preferences over groups of students under responsive preferences.

2.1 The formal model

We will assume that the rules of the market are:

1. Any student may enrol to a college if they both agree.
2. Any college may choose to keep one or more of its positions unfilled.
3. Any student may remain unmatched if he or she wishes.

Let C and S be two finite and disjoints sets, $C = \{c_1, \cdots, c_n\}$ called the set of colleges and $S = \{s_1, \cdots, s_m\}$ called the set of students. Each student $s \in S$ has preferences over the colleges and each college $c \in C$ has preferences over the students.

As in the marriage model, we will assume that these preferences are complete and transitive [1], so they may be represented by an ordered list as follows:

$$P(c) = \{s_1, s_2, c, s_3, \cdots\}$$

[1] A relation β is complete if for all $x, y \in X$, $x\beta y$ or $y\beta x$, and it is transitive if for all $x, y, x, \in X$, $x\beta y$ and $y\beta z$ implies that $x\beta z$

This particular list denotes that s_1 and s_2 are acceptable students but, college c prefers to leave a position unfilled before enrolling somebody else, that is, all other students are unacceptable. Similarly we can form a preference list for each student in the market.

We will denote $c_i >_s c_j$ to indicate that student s prefers college c_i to c_j, and $c_i \geq_s c_j$ to indicate that s prefers c_i at least as well as c_j. Similarly we can compare preferences of colleges lists.

College c is acceptable to student s if $c \geq_s s$, and student s is acceptable to college c if $s \geq_c c$.

As in the marriage problem, we will abbreviate preferences lists just including the acceptable alternatives.

Definition 2.1.1. *We call quota of college c a positive integer $q_c \in \mathbb{N}$ which indicates the number of positions offered by the college, namely the maximum number of positions it may fill. If we denote a particular college by c_i, its quota will be denoted by q_i.*

An outcome of the college admissions model is a bilateral matching between students and colleges such that each student is matched to at most one college and each college is matched to at most its quota of students. As in the marriage model, if some student has not been assigned to any college he or she will be self-matched, and a college that has some number of unfilled positions will be matched to itself in each of those positions.

Now, we are going to give a formal definition of an outcome.

Definition 2.1.2. *A matching μ is a map from the set $C \cup S$ into the set of unordered families [2] of elements of $C \cup S$ such that:*

- $|\mu(s)| = 1$ *for every student and* $\mu(s) = s$ *if* $\mu(s) \notin C$.
- $|\mu(c)| = q_c$ *for every college c and if the number of students in $\mu(c)$, say r, is less than q_c, then $\mu(c)$ contains $q_c - r$ copies of c.*
- $\mu(s) = c$ *if and only if* $s \in \mu(c)$.

Example. *We will represent a matching as:*

$$\mu = \begin{pmatrix} & c_1 & & c_2 & (s_4) \\ s_1 & s_3 & c_1 & s_2 & s_4 \end{pmatrix}$$

This matching represents that college c_1, which has a quota $q_1 = 3$, is matched with two students s_1 and s_3 and leaves an unfilled position. College c_2 has a quota $q_2 = 1$ and has been matched with s_2. Finally the student s_4 is unmatched.

At this point we clearly have a difference between the marriage model and the college admissions model. In the first one, we can compare matchings, that is, we could know which match would prefer each agent between different alternatives only comparing his or her own assignments in the different outcomes. In the college admission model, students will also compare matchings in the same way.

But, even thought we have described colleges' preferences over students, each college with a quota greater than one must be able to compare groups of student in order to compare

[2] For any set X, an unordered family of elements of X is a collection of elements, not necessarily distinct, in which the order is irrelevant.

alternative matchings. Hence we have to introduce a new concept that allows us to compare between groups of students.

Definition 2.1.3. *Let $P^{\#}(c)$ denote the preference relation of college c over all assignments $\mu(c)$ it could receive at some matching μ of the college admissions problem. A college c's preferences $P^{\#}(c)$ will be called responsive to its preferences $P(c)$ over individual students if, for any two assignments that differ in only one student, it prefers the assignment containing the most preferred student.*

We can state the definition formally as follows.

Definition 2.1.4. *The preference relation $P^{\#}(c)$ over the set of students is responsive if, whenever $\mu'(c) = \mu(c) \cup \{s_k\} \setminus \{\sigma\}$ for $\sigma \in \mu(c)$ and $s_k \notin \mu(c)$, then c prefers $\mu'(c)$ over $\mu(c)$ (under $P^{\#}(c)$) if and only if c prefers s_k to σ (under $P(c)$).*

It is clear by the definition that not all type of groups can be compared. For example if we suppose a college with a quota of two, responsiveness does not specify if this college prefers its first and fourth options instead of its second and third choices.

The assumption that colleges have responsive preferences is essentially no more than the assumption that their preferences for sets of students are related to their ranking of individual students in a natural way.

We will henceforth assume that colleges have preferences over groups of students that are responsive to their preferences over individual students as well as being complete and transitive, and that each agent's preferences over alternative matchings correspond exactly to his or her preferences over his or her own assignments at the two matchings.

2.2 Stability and group stability

As in the marriage model, our goal is to find how we can create stable matchings. For this reason we first need to define what is a stable matching and discuss what is group stability.

Definition 2.2.1. *A matching μ is individually irrational if $\mu(s) = c$ for some student s and a college c such that either the student is unacceptable to the college or the college is unacceptable to the student. Such a matching will also be said to be blocked by the unhappy agent.*

Definition 2.2.2. *A matching μ is blocked by the college-student pair (c, s) if $\mu(s) \neq c$ and $c >_s \mu(s)$ and $s >_c \sigma$ for some $\sigma \in \mu(c)$.*

From this last definition we note that σ may equal either some student s' or an unfilled position.

Now, we can define what a stable matching is in the sense discussed for the marriage model.

Definition 2.2.3. *A matching μ is stable if its not blocked by any individual agent or any college-student pair.*

In contrast to stable matchings in the marriage model, the definition of stability in the college admissions model is not so clear because a matching is one to many so we might consider coalitions consisting of a college and several students, or even coalitions consisting of multiple colleges and students. In what follows we shall prove that when preferences are responsive, nothing is lost by concentrating on simple college-student pairs.

Definition 2.2.4. *A matching μ is group unstable, or it is blocked by a coalition A, if there exists another matching μ' and a coalition A, which may consist of multiple students and/or colleges, such that for all students $s \in A$ and for all colleges $c \in A$:*

- $\mu'(s) \in A$, *i.e., every student in A who is matched by μ' is matched to a college in A.*
- $\mu'(s) >_s \mu(s)$, *i.e., every student in the coalition prefers matching μ' to the matching μ.*
- *If $\sigma \in \mu'(c)$ then $\sigma \in A \cup \mu(c)$, which means that every college in A is matched at μ' to new students only from A, although it can continue being matched with some of its old students from $\mu(c)$.*
- $\mu'(c) >_c \mu(c)$.

From this definition we can conclude that a matching μ is blocked by some coalition A if, by a matching among themselves, the students and colleges in A could all get an assignment preferable to μ.

Definition 2.2.5. *An outcome is a group stable matching if it is not blocked by any coalition.*

Now, we shall prove that, when preferences are responsive, this definition of group stability is equivalent to the definition we give for college-students pairs (definition 2.2.3).

Theorem 2.2.1. *Under responsive preferences a matching is group stable if and only if it is stable (by pairs).*

Proof. We shall prove both implications.

$\Rightarrow$ We suppose that μ is unstable via an individual agent (college or student) or via a college-student pair. Under this assumption μ is clearly unstable via the coalition consisting of this or these agents.

$\Leftarrow$ We suppose that μ is blocked via a coalition A and an outcome μ', this means $\mu'(s) >_s \mu(s)\ \forall s \in A$. Let $c \in A$ then, by the assumption, $\mu'(c) >_c \mu(c)$ so there exist a student $s \in \mu'(c) - \mu(c)$ and $\sigma \in \mu(c) - \mu'(c)$ such that $s >_c \sigma$.

If we suppose that $\sigma \geq_c s$, $\forall \sigma \in \mu(c) - \mu'(c)$, then $\mu(c) \geq \mu'(c)$, because preferences are responsive and transitive, which is a contradiction.

Hence, take that student $s \in \mu'(c) - \mu(c)$, there exists $\sigma \in \mu(c) - \mu'(c)$ and $s >_c \sigma$. Note that $s \in \mu'(c) - \mu(c)$ implies $s \in A$ and hence $c >_s \mu(s)$ so μ is unstable via s and c.

□

Note that this theorem achieves the same conclusion as the theorem in the marriage model which says that instabilities that can arise from coalitions of any size can be identified by examining only small coalitions.

2.3 College admissions model versus marriage model

Once analysed the notion of stability in the college admission model, we will investigate if we can guarantee the existence of stable matchings, as it is the case of the marriage model.

2.3.1 College admissions as an induced marriage market

The last theorem in the previous section, Theorem 2.2.1, allows us to concentrate in small coalitions instead of large groups of agents, so we identify stable matchings using only individual preferences, that is the list of preferences P, without knowing the preferences that a college has over a group of students $P^{\#}(c)$.

This suggest that the college admissions model may be very similar indeed to the marriage model. To describe this similarities we have to transform, a little, the model we described in section 2.1. Doing that, many results obtained for the marriage model will generalize immediately to the college admissions model.

Suppose a particular college admission problem such that there are n colleges $C = \{c_1, \cdots, c_n\}$ with quotas $q_1, \ldots, q_n$ and m students $S = \{s_1, \ldots, s_m\}$. Their preferences are given by the list $P = \{P(c_1), \ldots, P(c_n), P(s_1), \ldots, P(s_m)\}$.

We can suppose that each college c_i $\forall i = 1, \cdots n$ with a quota $q_i \geq 1$ is broken in q_i pieces of itself. Hence for, in our new market there will be students and college positions, each of them of quota one. That is, we replace c_i by q_i positions of c_i denoted by $c_{i,1}, \cdots, c_{i,q_i}$. Each of these positions have preferences over students that are identical to those of c_i.

We are going to assume that a student s for whom college c_j is acceptable will strictly prefer $c_{j,1}$ over all the other positions of c_j.

If preferences are strict, we have built a one-to-one correspondence between matchings in the original admissions problem and matchings in the marriage model.

If preferences are strict there only exists one marriage market that corresponds to the college admissions model but, if preferences over individuals are not strict (if colleges are indifferent between some students), there can be more than one matching in the related marriage market corresponding to a given matching of the college admissions problem. We will not consider this last option.

Theorem 2.3.1. *A matching of the college admissions problem is stable if and only if the corresponding matching of the related marriage market is stable.*

This theorem and the previous construction we did in this section allows us to conclude some results without proving them, simply considering them as an extension of the marriage problem.

Gale and Shapley in [4] observed that the algorithm discussed for the marriage problem, the deferred acceptance algorithm, could be modified for the college admission problem. This fact allows us to extend some theorems we have presented in previous sections that are directly consequences of the deferred acceptance algorithm. We are going to present these theorems giving two versions: the first one is from the marriage problem and the second one is its extension to the college admission problem.

Theorem 2.3.2. *Existence theorem.*

- *Marriage model: For any marriage market (M, W, P) there always exists at least one stable matching.*
- *College admission model: For any college admissions problem (C, S, P) there always exists at least one stable matching.*

Theorem 2.3.3. *Optimality theorem.*

- *Marriage model: When all men and women have strict preferences there always exists a M-optimal stable matching and a W-optimal stable matching.*
- *College admission model: When agents have strict preferences, the set of stable outcomes contains a C-optimal stable outcome and a S-optimal stable outcome.*

Although we have not defined optimality concepts for the college admission problem, due to the construction we have done, the definition must be clearly analogous in both markets.

2.3.2 Limitations to the relationship with the marriage market

In section 2.1, when we introduce our model, we see that one of the most important differences between both markets is that in many to one markets not all outcomes are comparable. That is, for a college $c \in C$ with preferences $P(c) = \{s_1, s_2, s_3, s_4\}$ the matching which assigns students s_2 and s_3 cannot be compared with the matching which assigns students s_1 and s_4.

The purpose of this section is to present one theorem which behaves differently in both markets, mainly because of the fact we have explained and even though colleges have responsive preferences. This illustrates that we cannot see the college admission model as a simple extension of the marriage market.

Theorem (Weak Pareto Optimality) There does not exist any outcome μ that every man prefers to the M-optimal stable matching μ_M in the marriage problem. Similarly, there exists no outcome μ' preferred by all women to μ_W.

Proposition 2.3.1. *When colleges have responsive preferences, the conclusion about Weak Pareto Optimality is false for the college admissions problem; there may exist outcomes that colleges strictly prefer to the C-optimal stable matching.*

Proof. The proof will be by means of an example.

Let us consider a market consisting of three colleges $C = \{c_1, c_2, c_3\}$ and four students $S = \{s_1, s_2, s_3, s_4\}$. College c_1 has a quota $q_1 = 2$ and both other colleges have a quota of 1. We are going to suppose that every college prefers some student before leaving a position unmatched. Figure 2.1 shows how their lists of preferences are:

The deferred acceptance algorithm with colleges choosing gives the C-optimal stable outcome:

$P(c_1)$	s_1	s_2	s_3	s_4	$P(s_1)$	c_3	c_1	c_2
$P(c_2)$	s_1	s_2	s_3	s_4	$P(s_2)$	c_2	c_1	c_3
$P(c_3)$	s_3	s_1	s_2	s_4	$P(s_3)$	c_1	c_3	c_2
					$P(s_4)$	c_1	c_3	c_2
Colleges preferences					Students preferences			

Figure 2.1: Preference list on a college admissions problem.

$$\mu_C = \begin{pmatrix} c_1 & c_2 & c_3 \\ s_3\ s_4 & s_2 & s_1 \end{pmatrix}$$

If we consider now the feasible outcome μ such that:

$$\mu = \begin{pmatrix} c_1 & c_2 & c_3 \\ s_2\ s_4 & s_1 & s_3 \end{pmatrix}$$

we clearly can see that outcome μ gives to c_2 and c_3 their first choice which is preferred by them over their assignment in μ_C which match them with their second choice. Outcome μ is also preferred than outcome μ_C for c_1, because preferences are responsive and μ_C gives to c_1 its third and fourth choice students and outcome μ gives to c_1 its second and fourth options which are better than the other ones due to the definition of responsive preferences.

This completes the demonstration that the conclusion of Weak Pareto Optimality is false in the college admissions problem when colleges have responsive preferences. □

2.4 The labour market for medical interns

We are going to present here a well known real-life application of the college admission model.

The National Resident Match Program (NRMP) [3], began in 1952 in response to dissatisfaction with the process and results of matching applicants to residency programs via the decentralized, competitive market. From shortly after the first residency programs were formally introduced, the hiring process was "characterized by intense competition among hospitals for (an inadequate supply) of interns." In general, hospitals benefited from filling their positions as early as possible, and applicants benefited from delaying acceptance of positions. The combination of these factors lead to offers being made for positions up to two years in advance. While efforts made to delay the start of the application process were somewhat effective, they ultimately resulted in very short deadlines for responses by applicants, and the opportunities for dissatisfaction on the part of both applicants and hospitals remained. The students in 1951 protested against the originally proposed matching algorithm, and objected to the hospital-optimal nature of the proposed algorithm. The

[3] In 1968 the National Intern Matching Program (NIMP) was renamed the National Intern and Resident Matching Program (NIRMP), and in 1978 renamed the National Resident Matching program (NRMP) to reflect the changes suffered in the structure of positional graduate medical training.

NRMP stated that students from 1951 objected to an algorithm that gave them incentives to misrepresent their true preferences.

A publication in 1962 by Gale and Shapley noted that there always exists a stable solution when colleges are matched to students, but that it is possible to favour colleges as a group over applicants as a group (and vice-versa). That is, Gale and Shapley found that there is a college-optimal stable matching and an applicant-optimal stable matching.

Controversy arose regarding whether the program was susceptible to manipulation or unreasonably fair to employers. Indeed, it was shown that in simple cases (i.e. those that exclude couples, second-year programs, and special cases for handling unfilled slots) that had multiple "stable" matchings, the algorithm would return the solution that was best for the hospitals and worst for the applicants. It was also susceptible to collusion on the part of hospitals: if hospitals were to organize their preference lists properly, the result returned would be completely unaffected of the preference lists of the residents. A correspondence in New England Journal of Medicine in 1981 recognized that the algorithm in use was hospital-optimal for individual applicants, in direct contradiction to the NRMP's published statements.The promotional NRMP literature was revised to remove the detailed, step-by-step description of their algorithm that had been there before.

Despite many indications for updating the NRMP algorithm, it saw only minor and incremental changes after its institution in 1952 until 1997. However, in the fall of 1995 the Board of Directors of the NRMP commissioned a preliminary research program for the evaluation of the current algorithm and of changes to be considered in its operation and description, and a study comparing a new algorithm with the existing one. The new algorithm was adopted in May 1997 and has been in use since its first application in March 1998, although the study showed that the net effect of the change on actual matches has been minimal.

Our main object in what follows is deeply study the first version of the assignment algorithm established by NIMP. As we explained when we introduced this discussion, before 1945 there were a lot of reasons for continually advancing the date at which interns were appointed. Although hospitals all preferred as late an appointment as possible, each preferred to appoint its interns earlier than its competitors: the situation is well modelled as a multiple-agent Prisoner's Dilemma [4].

To analyse the market after 1945 we consider a market formed by a set $H = \{h_1, \cdots, h_m\}$ of hospital programs offering positions to first-year graduates and a set $S = \{s_1, \cdots, s_n\}$ of graduating students. The rules of the game are those one we explained at the college admission model. Actually, the labour market for medical interns is an example of a college admissions problem where hospitals act as colleges and medical interns as students.

We are going to present the NIMP algorithm, extracted literally from [9].

> Each hospital program rank orders the students who have applied to it (marking "X" any students who are unacceptable) and each

[4] A Prisoner's Dilema is a paradox in decision analysis in which two individuals acting in their own best interest pursue a course of action that does not result in the ideal outcome. The typical prisoner's dilemma is set up in such a way that both parties choose to protect themselves at the expense of the other participant. As a result of following a purely logical thought process to help oneself, both participants find themselves in a worse state than if they had cooperated with each other in the decision-making process.

student rank orders the hospital programs to which he has applied (similarly indicating any which are unacceptable). These lists are mailed to the central clearinghouse, where they are edited by removing from each hospital program's rank-order list any student who has marked that program as unacceptable, and by removing from each student's list any hospital which has indicated he is unacceptable. The edited lists are thus rank orderings of acceptable alternatives.

These lists are entered into what may be though of as a list-processing algorithm consisting of a matching phase and a tentative-assignment-and-update phase. The first step of the matching phase (the 1:1 step) checks to see if there are any students and hospital programs which are top-ranked in one another's ranking. (If a hospital h_i has a quota of q_i then the q_i highest students in its ranking are top-ranked.) If no such matches are found, the matching phase proceeds to the 2:1 step, at which the second ranked hospital program on each student's ranking is compared with the top-ranked students on that hospital's ranking. At any step when no matches are found, the algorithm proceeds to the next step, so the generic k:1 step of the matching phase seeks to find student-hospital pairs such that the student is top-ranked on the hospital's ranking and the hospital is kth ranked by the student. At any step where such matches are found, the algorithm proceeds to the tentative-assignment-and-update-phase.

When the algorithm enters the tentative-assignment-and-update-phase from the k:1 step of the matching phase, then the k:1 matches are tentatively made; i.e., each student who is a top-ranked choice of his kth choice hospital is tentatively assigned to that hospital. The rankings of the students and hospitals are then updated in the following way.

Any hospital which a student s_j ranks lower than his tentative assignment is deleted from his ranking (so the updated ranking s_j tentatively assigned to his kth choice now lists only his first k choices) and student s_j is deleted from the ranking of any hospital which was deleted from s_j's ranking (so the updated rankings of each hospital now include only those applicants who haven't yet been tentatively assigned to a hospital they prefer).

Note that, if one of a hospital's top-ranked candidates is deleted from its ranking, then a lower-ranked choice moves into top-ranked category, since the hospital's updated ranking has fewer students, but the same quota, as its original ranking. When the rankings have been updated in this way, the algorithm returns to the start of the matching phase, which examines the updated rankings for new matches. Any new tentative matches found in the matching phase replace prior tentative matches involving the same student.

> (Note that new tentative matches can only improve a student's tentative assignment, since all lower ranked hospitals have been deleted from his ranking.)
>
> The algorithm terminates when no new tentative matches are found, at which point tentative matches become final. That is, the algorithm matches students with the hospitals to which they are tentatively matched when the algorithm terminates. any student or hospital position which was not tentatively matched during the algorithm is left unassigned, and must make subsequent arrangements by directly negotiating with other unmatched students or hospitals.

Theorem 2.4.1. *The NIMP algorithm is a stable matching.*

Proof. When the algorithm ends, each hospital h_i is matched with the top q_i choices on its final updated rank-order list. This assignment is stable, since any student s_j who some hospital h_i originally ranked higher than one of its final assignees was deleted from h_i's ranking when s_j was given a tentative assignment higher in his or her ranking than h_i. Hence, the final assignment give s_j a position he or she ranked higher than h_i. So the final matching is not unstable with respect to any such (h_i, s_j). □

The fact of stability explains why the NIMP algorithm was able to achieve such rates of voluntary participation.

Theorem 2.4.2. *For any submitted lists of (strict) preferences over individuals, the NIMP algorithm produces a matching that gives each hospital h_i its ranked achievable students.*

This theorem shows that the algorithm implemented by NIMP is hospital-optimal but if hospitals act as students and vice versa the algorithm is student-optimal. As we have shown in chapter one, there does not exist a stable M-optimal and W-optimal matching simultaneously (unless only one stable matching exists) so, depending of who is proposing the outcome turns into a sense or into another sense.

In the college admissions model we have the same problem and due to the fact that NIMP published the algorithm they used a new controversy arise: NIMP decided to do the algorithm in terms of optimality for hospitals and students completely disagree.

Chapter 3

The school choice model

School choice is one of the widely discussed topics in education. It means giving parents the opportunity to choose the school their child will attend.

The main purpose of this chapter is to present the complexity of school choice problem as a two-sided matching assignment problem. We are going to study children assignments at US cities (mainly Boston and New York), which were the cities where matching theory was first applied to solve the assignment problems.

Traditionally, children were assigned to public schools according to where they live. Wealthy parents already have the possibility of choosing school, because they can afford to move to an area with good schools, or they can enrol their child in a private school. Parents without such means, until recently, had no choice of school, and had to send their children to schools assigned to them by the district, regardless of the school quality or appropriateness for the children.

In real-life many of school assignment plans had protocols and guidelines for the student assignment without explicit procedure. In spite of that, in some cities school choice programs are accompanied with explicit procedures, but parents could do strategic behaviours to reach their first school option. For these reasons a central issue is created in order to design a fair mechanism to assign students, since the main problem is the impossibility to assign each student to his top choice school.

Abdulkadiroglu and Sönmez, [1] and [2], and Gale [4] proposed three different ways to assign students in a fair way.

In the two previous chapters we study marriage market and college admission market, by exploring the kind of matchings we might expect to observe and its properties. In school choice problem a concept even more important than stability appears, that is strategic behaviour, which is the study of how we should expect agents to behave.

The goal of this work is not to issue a verdict about which of these procedures gives a better chance to reach equality between children, it is only to present the alternatives and study their properties in such a way they give us tools to decide which of them we want to apply to make the assignment. Nevertheless this section presents the reasonable doubt about if it really exists a procedure that gives all children equal opportunities.

3.1 The formal model

Let be I and S to disjoint sets, $I = \{i_1, \dots, i_n\}$ will be the set of students and $S = \{s_1, \dots, s_m\}$ the set of schools. As in Chapter 2, we are going to consider a capacity vector (a quota for each school), as $q = (q_{s_1}, \dots, q_{s_m})$. We also are going to consider a profile of strict preferences for students $P = (p_{i_1}, \dots, p_{i_n})$ and a strict priority structure of the schools over students $f = (f_{s_1}, \dots, f_{s_n})$. That is, a school choice problem is a 5-tuple (I, S, q, P, f).

We denote i and s a a generic student and a generic school, respectively. The preference relation P_i has the same notation that in the previous models, so, $P_i \in S \cup \{i\}$, where, if $P_i = \{i\}$ then this students is outside the market (for example enrolling on a private school). If we denote $s' >_i s$, then student i prefers school s' over s and finally, school s is acceptable to i if $s >_i i$.

The priority ordering f_s of school s assigns ranks to the students according to their priority for school s. The rank of student i for school s is $f_s(i)$. Then, $f_s(i) < f_s(j)$ means that student i has higher priority (or lower rank) for school s than student j.

This priority structure of the schools over students is the main difference between the college admission model and the school choice model. We can observe that in school choice model schools are mere objects to be consumed by students, whereas in the college admission model both sides of the market are agents with preferences over the other side.

Definition 3.1.1. *In a particular school choice problem (I, S, q, P, f) an outcome of this problem is a function μ from the set $S \cup I$ into itself such that:*

- $\forall i \in I$, $|\mu(i)| = 1$ *and* $\mu(i) \in S \cup \{i\}$.
- $\forall s \in S$, $|\mu(s)| \leq q_s$ *and* $|\mu(s)| < q_s$ *if and only if all students are assigned. Schools cannot leave voluntarily a position unfilled.*
- $\forall s \in S$ *and* $i \in I$, $\mu(i) = s$ *if and only if* $i \in \mu(s)$.

As we have seen in chapter 1 and 2, stability plays a crucial role in matching problems, and this is also the case for school choice problems.

Definition 3.1.2. *Let be (I, S, q, P, f) a school choice problem. A matching μ is stable if:*

- *It is individually rational, i.e.,* $\forall i \in I$, $\mu(i) >_i i$.
- *Is is non wasteful, i.e., for all* $i \in I$ *and* $\forall s \in S$, $s >_i \mu(i)$ *implies* $|\mu(s)| = q_s$.
- *There is no justified envy, there is no blocking pair, i.e., for all* $i, j \in I$ *with* $\mu(j) = s$, $s >_i \mu(i)$ *implies* $f_s(j) < f_s(i)$.

In school choice, stability can be understood as a fairness criterion since no children will attend school that is not acceptable for him and there is no situation in which student i is matched to a worse school than school s, if s admits another student who has lower priority at s than i does (or has a vacant seat).

Weak Pareto optimality is defined by the same way as we defined it in chapter one. The only difference is that we only study optimality for students, since schools have not preferences over students.

3.2 Strategic questions

When we introduce this section, we mentioned the importance to predict how agents can act if they know how the assignment works. The hypothesis of knowledge about how the mechanism works is essential in what follows, otherwise this study has no sense.

The study of agents behaviour is known as strategic questions. To address these questions of individual behaviour, we need to model the decisions that individuals may be called upon to make in the course of the market. The main question when we analyse strategic behaviour can be formulated as:

It is always in each agent's best interest to state his or her true preferences to the matchmaker?

The answer of this question is No, there will be situations in which some agent could obtain a preferred partner (or school or college) if he behaves strategically.

We need to introduce some notation in order to develop this new theory. If Q represents the choices of all agents, and we want to focus on the decision facing one of them, say agent i, then we will write $Q = (Q_i, Q(i))$, where $Q(i)$ is the choice of player i and Q_i is the set of choices of all agents other than i. If we consider $Q' = (Q_i, Q'(i))$, then Q' and Q differ only in player i's choice.

Definition 3.2.1. *A matching mechanism is a function h whose range is the set of all possible inputs, and whose output is a matching $h(Q)$.*

Definition 3.2.2. *A particular strategy choice $Q^*(i)$ by a player i is a best response by i to Q_i if player i likes the outcome $h(Q_i, Q^*(i))$ at least as well as any of the outcomes $h(Q_i, Q(i))$ that is, $h(Q_i, Q^*(i)) \geq_i h(Q_i, Q(i))$.*

Definition 3.2.3. *A dominant strategy for an agent i is a strategy $Q^*(i)$ that is the best response to any Q_i of the remaining agents.*

If $h(Q)$ is always stable with respect to Q, it will be called a stable matching mechanism. Similarly if $h(Q)$ is always Pareto optimal with respect to set of children (or agents in general if we are studying another model), then it will be called a Pareto optimal stable matching mechanism.

Definition 3.2.4. *A matching mechanism will be called strategy proof if it makes it a dominant strategy for each player to state his true preferences in the strategic game it induces.*

The aim in what follows is to know if optimality, stability and strategy-proofness can be given simultaneously in an arbitrary matching mechanism.

Theorem 3.2.1. ***Impossibility Theorem*** *No stable matching mechanism exists for which stating the true preferences is a dominant strategy for every agent.*

Proof. It will be sufficient to demonstrate that a matching problem exists for which no stable matching procedure has truthful revelation as a dominant strategy. Consider a market with two agents in both sides of the market $\{i_1, i_2\}$ and $\{j_1, j_2\}$, i.e. a one-to-one market, with preferences P given by $P(i_1) = \{j_1, j_2\}$, $P(i_2) = \{j_2, j_1\}$, $P(j_1) = \{i_2, i_1\}$ and $P(j_2) = \{i_1, i_2\}$.

Then there are two stable matchings μ and ν given by: $\mu(i_k) = j_k$ for $k = 1, 2$ and $\mu(i_k) = j_l$ for $k, l = 1, 2$ and $k \neq l$. So any stable mechanism must choose either μ or ν when preferences P are stated.

Suppose the mechanism choose μ. We can observe, for example, that if j_2 changes his stated preferences from $P(j_2)$ to $Q(j_2) = i_1$ while everyone else states their true preferences, then ν is the only stable matching with respect to the preferences $P' = (P(i_1), P(i_2), P(j_1), Q(j_2))$, and any stable mechanism must select ν when the stated preferences are P'.

So it is not a dominant strategy for all agents to state their true preferences, note that j_2 improves by misrepresenting his preferences. That is all stable matching mechanism are not strategy proof. □

Theorem 3.2.2. *The mechanism that yields a children optimal stable matching (in terms of stated preferences) makes it a dominant strategy for each child to state his true preferences.*

Proof. For a detailed proof see [8] □

3.3 Matching mechanisms

In this section we are going to present different assignment algorithms to form matchings given a school choice problem. The main difference between them is the way the students are rejected.

In general [1], for each school a priority ordering is determined according to the following hierarchy:

- First priority: sibling and walk zone.
- Second priority: sibling.
- Third priority: walk zone.
- Fourth priority: other students.

For each algorithm we are going to give a practical example. When the algorithm would be described we are going to study properties of the resulting matching. These last examples will show the importance of the design of the algorithms in school choice, since each algorithm we are going to present gives a different matching with completely different properties.

[1]Catalonia assigned accumulative points to students according to the following hierarchy: sibling (40 points), home walk zone (30 points), parents work walk zone (20 points), school and home in the same town (10 points), disabilities (10 points) and complementary points for poor economic situations maximum of 15 points depending on the situation.

And at this point, matching theory cannot decide which mechanism is better. It is then time for authorities to choose which property is more desirable: stability or strategy-proof?

3.3.1 Boston mechanism

Boston mechanism is known as an immediate acceptance algorithm. Let us explain the main ideas of this algorithm, based on [1].

Step 1: Each student submits a preference ranking of the schools.

Step 2: In Step 1 only the first choices of the students are considered. For each school, consider the students who have listed it as their first choice and assign seats of the school to these students one at a time following their priority order until either there are no seats left or there is no student left who has listed it as his first choice.

Step k: Consider the remaining students. All students who are assigned in a previous step cannot be removed. In Step k only the kth choices of these students are considered. For each school still with available seats, consider the students who have listed it as their kth choice and assign the remaining seats to these students one at a time following their priority order until either there are no seats left or there is no student left who has listed it as his kth choice.

The procedure terminates after any step k when every student is assigned a seat at a school, or if the only students who remain unassigned listed no more than k options.

Example. *We are going to apply the Boston mechanism to a small market in order to show that it can produce unstable matchings. Let be $I = \{i_1, i_2, i_3\}$ the set of children and let be $S = \{s_1, s_2\}$ the set of schools. Consider the preferences profile detailed in 3.1, where each school has a quota of one:*

$P(i_1)$	s_2	s_1	$P(s_1)$	i_1	i_2	i_3
$P(i_2)$	s_1		$P(s_2)$	i_3	i_1	
$P(i_3)$	s_1	s_2				
Children preferences			School preferences			

Figure 3.1: Preference list on a marriage market

The resulting matching applying Boston mechanism is:

$$\mu = ((i_1, s_2), (i_2, s_1), (i_3, \emptyset))$$

Clearly the resulting matching is not stable since pair (i_3, s_2) is a blocking pair and it is not strategy-proof since children i_3 could gain presenting an alternative preference relation $P'(i_3) = s_2$.

The Boston mechanism attempts to assign as many students as possible to their first choice school, and only after all such assignments have been made does it consider assignments of students to their second choices, etc. The main problem in this assignment mechanism is that: if a student is not admitted to his first choice school, his second choice may already be filled with students who listed it as their first choice. That is, a student may fail to get a place in his second choice school that would have been available had he listed that school as his first choice. This has the potential both to change the preference rankings that some families submit, and to work to the disadvantage of families that fail to take into account of such strategic considerations.

Proposition 3.3.1. *The Boston mechanism is not strategy-proof and can produce unstable matchings.*

Definition 3.3.1. *A school is over-demanded if the number of students who rank that school in his first choice is greater than the quota of this school.*

Proposition 3.3.2. *No one who lists an over-demanded school as a second choice will be assigned to it by the Boston mechanism, and listing an over-demanded school as a second choice can only reduce the probability of receiving schools ranked lower.*

Boston mechanism was working since 2003 but finally it was replaced by a strategy-proof mechanism indeed to help families by allowing them to state their true preferences.

3.3.2 Top trading cycles mechanism

As already mentioned, it is costly under the Boston mechanism to list a first choice that you will do not succeed in getting because, once other students are assigned their first choice places, they cannot be displaced even by a student with higher priority. This is avoided under the student-proposing deferred acceptance algorithm. For a given list of preferences, student priorities and school capacities, this mechanism determines a student assignment following the same steps we described in previous sections for the marriage model or college admission model.

In contrast with the Boston algorithm, the deferred acceptance algorithm assigns seats only tentatively at each step, so students with higher priorities may be considered in subsequent steps. For these reason we have proved in previous sections that the resulting matching is stable. Moreover all students prefer their outcome to any other stable matching and the induced mechanism is strategy proof. If the intention of the school board is that priorities be strictly enforced this mechanism will be leading candidate.

However, if welfare considerations apply only to students, there is tension between stability and Pareto optimality. If priorities are merely a device for allocating scarce spaces, it might be possible to assign students to schools they prefer by allowing them to trade their priority at one school with a student who has priority at a school they prefer. The mechanism we want to analyse, the top trading cycles mechanism creates a virtual exchange for priorities. For a given list of priorities, student preferences and school capacities this mechanism determines a student assignment with the algorithm we are going to present.

Definition 3.3.2. *In the school choice model, a cycle is an ordered list of schools and students* $\{i_1, s_1, i_2, s_2, \ldots, i_k, s_k\}$ *with student* 1 *pointing to school* 1*, school* 1 *to student* 2*, ..., student* k *to school* k*, and school* k *pointing to student* 1.

Step 1: Assign counters for each school to track how many seats remain available. Each student points to his favourite school and each school points to the student with the highest priority. There must be at least one cycle. Each student is part of at most one cycle. Then each student in the cycle is assigned to the school she/he points to. The counter of each school is reduced by one and if it reaches zero, school is removed.

Step k: Each remaining student points to his favourite school among the remaining schools and each remaining school points to the student with the highest priority among the remaining students. There is at least one cycle. Every student in a cycle is assigned a seat at the school he points to and is removed. The counter of each school in a cycle is reduced by one and if it reaches zero, the school is removed.

The procedure terminates when each student is assigned a seat or all submitted choices have been considered. This algorithm due to [1] is an extension of Gale's top trading cycles described in [13].

Example. *Let be $I = \{i_1, i_2, i_3, i_4, i_5, i_6, i_7\}$ and $S = \{s_1, s_2, s_3\}$ with the preference profile detailed in figure 3.2 and 3.3. We are going to apply the algorithm detailed to form a matching.*

$P(i_1)$	s_1	s_2	s_3
$P(i_2)$	s_1	s_2	s_3
$P(i_3)$	s_2	s_3	s_1
$P(i_4)$	s_3	s_2	s_1
$P(i_5)$	s_2	s_3	s_1
$P(i_6)$	s_3	s_1	s_2
$P(i_7)$	s_2	s_1	s_3

Figure 3.2: Children's preference list

$P(s_1)$	$q_1 = 2$	i_1	i_2	i_3	i_4	i_5	i_6	i_7
$P(s_2)$	$q_2 = 2$	i_2	i_4	i_6	i_1	i_3	i_5	i_7
$P(s_3)$	$q_3 = 3$	i_1	i_3	i_5	i_7	i_2	i_4	i_6

Figure 3.3: School's preference list

1. *In this first step all students points to his favourite school and each school to his favourite student. If there are any cycles, we remove the student and we substract one to the count of the school. If the count is zero, we remove this school.*

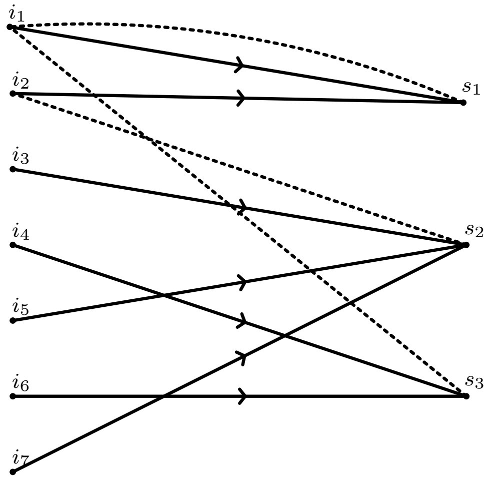

Figure 3.4: First step TTC

We can see (3.4) that we have one cycle, thus formed by pair (i_1, s_1), so we substract one in s_1's counter and we remove student i_1.

2. *In the second step all remaining students point to his most preferred school and each school points to his most preferred student remaining on the market.*

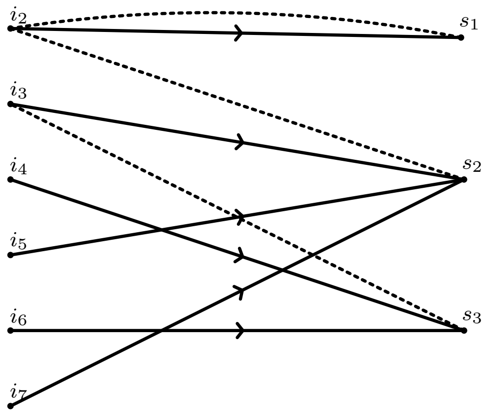

Figure 3.5: Second step TTC

The only cycle formed on this step (see 3.5) is that where (i_2, s_1) are matched. We have to substract one in the counter of s_1 and then this school has no more seats, so we remove it.

3. *In the third step we have a cycle formed by $(i_3, s_2, i_4, s_3, i_3)$ (see 3.6), that is we match i_3 to s_2 and i_4 to s_3.*

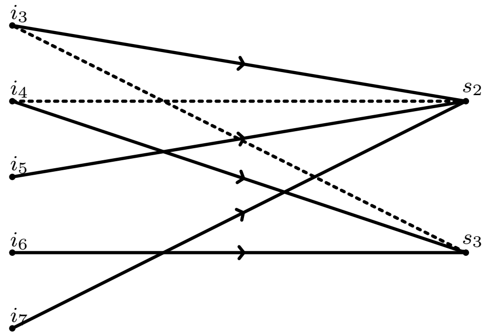

Figure 3.6: Third step TTC

4. *In this fourth step we have the graph described in 3.7. At this moment the counter of s_2 is one and two for s_3.*

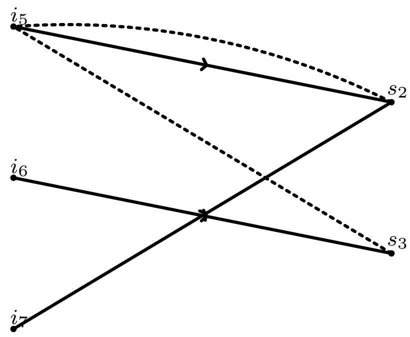

Figure 3.7: Fourth and last step TTC

After this step, the counter for school two is zero since (i_5, s_2) is a cycle. For this reason children i_6 and i_7 are automatically assigned to school s_3 where two positions are available.

The resulting matching is:

$$\mu = ((i_1, s_1), (i_2, s_1), (i_3, s_2), (i_4, s_3), (i_5, s_2), (i_6, s_3), (i_7, s_3))$$

Theorem 3.3.1. *The Top trading cycles mechanism is Pareto-effcient and strategy-proof.*

Proof. For a detailed proof see [1]. □

One of the most relevant properties when we analysed a matching is stability. We are going to see in the following example that the resulting matching of the top trading cycles algorithm may not be stable.

Example. *We are going to apply the top trading cycles algorithm to the same market we have applied the Boston mechanism in section 3.3.1. We are going to show that for this same market top trading cycles algorithm gives a completely different resulting matching.*

Let be $I = \{i_1, i_2, i_3\}$ *and* $S = \{s_1, s_2\}$ *with the preference list detailed in Figure 3.3.*

The resulting matching applying the top trading cycles mechanism is:

$$\mu = ((i_1, s_2), (i_2, \emptyset), (i_3, s_1))$$

We can observe that his resulting matching is not stable since the pair (i_2, s_1) *is a blocking pair but it is Pareto efficient and strategy proof.*

In appendix B we present a code in C that can produce the matching resulting of applying the top trading cycles procedure introducing three new assumptions, in order to simplify the problem.

Finally and concluding this chapter on school choice, we remark that three mechanisms are proposed. The Boston mechanisms is not stable and not strategy-proof, the Deferred Acceptance mechanism is stable and strategy-proof but does not guarantee efficiency. Compared to that, the Top Trading Cycles algorithm is efficient and strategy-proof but does not guarantee stability. The last two are strategy-proof, so they both avoid strategic behaviour on the side of parents.

Conclusions

Even if considering that the main objectives of this study have been achieved, as it allowed me to deepen in a rigorous way on the matching theory, I would have liked to study in a more detailed way the school choice model, and consequently be able to assess more accurately some issues that might seem incomplete in this report. A deeper assessment on the school choice model was not possible since it was first necessary to completely develop the one-to-one matching theory and the college admission model. Hence, a deeper analysis of the third model would have considerably enlarged the scope and amount of work of the present study.

One of the main and initial purposes of this final degree project was to complete it by studying the assignment model for all the Catalan primary schools. I have not been able to provide any results on this subject due to the lack of cooperation from the responsible authorities on this area (Catalan government and local authorities) who denied to provide any information related to school assignment due to the confidentiality level of this data. The impossibility of having the information from the authorities brought me to keep inquiring on this subject until very recently. Even not having the data about the Catalan school assignment model, many questions can be raised only by reading the rules that manage this matching mechanism. The main question is why parents can only apply for five schools, taking into account that, if their child is not selected in any of their choices, he enters into a completely random phase where he can be assigned to any school of the municipality (even very far away from his residence). The mentioned restriction seems illogical once the three assignment methods have been assessed.

The latest available references, mainly coming from the USA, rely into considering an assignment model in more quantifiable terms (stability, strategy, efficiency). In my opinion, even if this trend shall be strongly considered, it still has to be questioned the point-based system that determines the priority level of each child in each school. This system classifies children in terms of the points they get for each school. The points a child gets depend on his particular familiar, financial and social situation and will determine the school he is assigned to, trying to provide all the children with equal opportunities. In this study, no debate on the limitations that the model shows has been done as it was not its purpose, even though, it is clear that social conflicts might emerge as consequence of the limitations that the theoretical model shows. Regarding the situation exposed, I strongly expect that dedicated studies can analyse the Catalan situation of school assignments in a near future.

Finally, and despite all the difficulties that I had to face, the present study has given to me the opportunity to see a practical application on real life matters of different subjects I have learned during my years of studies.

Appendix A

Deferred acceptance C code

In chapter one we have analysed the deferred acceptance algorithm and how it can give an M-optimal stable matching if men are proposing or an W-optimal stable matching if are women who are proposing.

In this appendix we write a code, in language programming C, that will return to the user the M-optimal stable matching if the market given is square, that is, if there are as many men as women and all agents in one side of the market are acceptable for the agents on the other side of the market.

```
/*Deferred acceptance algorithm*/

#include<stdio.h>
#include<stdlib.h>

int main (void){
        int n, **m, **w, *a, *msin,*mpref, i, j, k, cont, m1, m2;

/*Dinamic memory for matrix*/
        printf("Deferred acceptance algorithm, men proposing\n");
        printf("Size of the market, n = ");
        scanf("%d", &n);

        m = (int**)malloc(n*sizeof(int*));
        for(i=0; i<n; i++){
                m[i] = (int*)malloc(n*sizeof(int));
                if(m[i]==NULL){
                        puts("malloc error");
                        exit(1);
                }
        }

        w = (int**)malloc(n*sizeof(int*));
        for(i=0; i<n; i++){
```

```
                w[i] = (int*)malloc(n*sizeof(int));
                if(w[i]==NULL){
                        puts("malloc error");
                        exit(1);
                }
        }

        a = (int*)malloc(n*sizeof(int));
        if(a==NULL){
                puts("malloc error");
                exit(1);
        }

        msin = (int*)malloc(n*sizeof(int));
        if(msin==NULL){
                puts("malloc error");
                exit(1);
        }

        mpref = (int*)calloc(n, sizeof(int));
        if (mpref==NULL){
                puts("calloc error");
                exit(1);
        }

/*Reading data*/
        printf("Men list of preferences:\n");
        for(i=0; i<n; i++){
                for(j=0; j<n; j++){
                        scanf("%d", &m[i][j]);
                }
        }

        printf("Women list of preferences:\n");
                for(i=0; i<n; i++){
                        for(j=0; j<n; j++){
                                scanf("%d", &w[i][j]);
                        }
                }

/*Initialize vectors*/
        for(i=0; i<n; i++){
                a[i] = -1;
                msin[i] = -1;
        }
```

```
/*DA algorithm*/
        cont = n;
        while(cont!=0){
                for(i=0; i<n; i++){
                        if(msin[i]==-1){
                                k = m[i][mpref[i]];
                                /*Match man i with woman k*/
                                if(a[k] == -1){
                                        a[k] = i;
                                        msin[i] = 0;
                                }else{
                                        m1 = -1;
                                        m2 = -1;

                                        for(j=0; j<n; j++){
                                          if(w[k][j]==a[k])
                                                m1 = j;
                                          if(w[k][j]==i)
                                                m2 = j;
                                          if(m1!=-1 && m2!=-1)
                                                break;
                                        }

                                        if(m2<m1){
                                          msin[a[k]]=-1;
                                          mpref[a[k]]++;
                                          a[k] = i;
                                          msin[i] = 0;

                                        }else{
                                          mpref[i]++;
                                        }
                                }
                        }
                }
                cont = 0;
                for(i=0; i<n; i++){
                        if(msin[i]==-1) cont++;
                }
        }

/*Solution*/
        for(i=0; i<n; i++){
                printf("(m%d,w%d)\n", a[i], i);
        }
        free(a);
```

```
	for(i=0; i<n; i++){
		free(w[i]);
		free(m[i]);
	}
	free(w);
	free(m);
	free(mpref);

	return 0;
}
```

Appendix B

Top Trading Cycles C code

In chapter three we have analysed the Top Trading Cycles algorithm to assign children to schools. Due to its importance in front of Boston mechanism, in this appendix we present a code, in language programming C, that will return the matching resulting of applying the algorithm described.

In order to simplify the problem we add some assumptions:

- No child can remain unmatched.
- Agents in both side of the market have a complete list of preferences.
- There are at least as many seats as children.

```
/*Top Trading Cycles algorithm*/

#include<stdio.h>
#include<stdlib.h>

int solution (int*, int**, int*, int*, int, int, int, int);
int cycle (int*, int, int);
void fillsol(int**, int, int);

int main (void){
        int i, j, k, cont, ag, seats, ns, nc, **sh, **ch, *ch1,
           **sol, *qs, *v, cy, child, school;

/*Dinamic memory for matrix*/
        printf("Top trading cylces algorithm with more seats than
           students\n");
        printf("Number of children:\n");
        scanf("%d", &nc);
        printf("Number of schools:\n");
        scanf("%d", &ns);
```

```
ch = (int**)malloc(nc*sizeof(int*));
for(i=0; i<nc; i++){
        ch[i] = (int*)malloc(ns*sizeof(int));
        if(ch[i] == NULL){
                puts("Mallor error");
                exit(1);
        }
}

sh = (int**)malloc(ns*sizeof(int*));
for(i=0; i<ns; i++){
        sh[i] = (int*)malloc((nc)*sizeof(int));
        if(sh[i] == NULL){
                puts("Mallor error");
                exit(1);
        }
}

v = (int*)malloc((ns+nc+1)*sizeof(int));
if(v == NULL){
        puts("Calloc error");
        exit(1);
}

ch1 = (int*)malloc((nc)*sizeof(int));
if(ch1 == NULL){
        puts("Malloc error");
        exit(1);
}

sol = (int**)malloc((ns)*sizeof(int*));
for(i=0; i<ns; i++){
        sol[i] = (int*)malloc(nc*sizeof(int));
        if(sol[i] == NULL){
                puts("Malloc error");
                exit(1);
        }
        for (j = 0; j < nc; j++)
                sol[i][j] = -1;
}


qs = (int*)malloc(ns*sizeof(int));
if(qs == NULL){
        puts("Malloc error");
        exit(1);
}
```

```
/*Reading data*/
	printf("Children list of preferences over all possible 
	   schools:\n");
	for(i=0; i<nc; i++){
		ch1[i] = i;
		for(j=0; j<ns; j++){
			scanf("%d", &ch[i][j]);
			puts("Error");
		}
	}

	printf("Schools quota:\n");
	for(i=0; i<ns; i++){
		scanf("%d", &qs[i]);
		seats += qs[i];
	}
	if(seats<nc){
		puts("ERROR: At least many chairs as students are
		   required");
		exit(2);
	}

	printf("Schools priority over all possible students:\n");
	for(i=0; i<ns; i++){
		for(j=0; j<nc; j++){
			scanf("%d", &sh[i][j]);
		}
	}
	printf("\nChildren\n-------------- \n");
	for(i=0; i<nc; i++){
		printf("Child%d: ", i);
		for(j=0; j<ns; j++){
			printf(" %d ", ch[i][j]);
		}
		printf("\n");
	}

	printf("\nSchools\n--------------\n");
	for(i=0; i<ns; i++){
			printf("School%d:", i);
			printf(" q%d = %d || ", i, qs[i]);
		for(j=0; j<nc; j++){
			printf(" %d ", sh[i][j]);
		}
```

```
                printf("\n");
        }

/*Top trading cycles algorithm*/
        cont = nc;
        j = 0;
        while(cont!=0){
                for(i=0; i<(ns+nc+1); i++)
                        v[i] = -1;
                /*Fill the vector where we are going to find a
                    cycle*/
                while(ch1[j]==-1)
                        j++;
                v[0] = ch1[j];
                ag = ch1[j];
                i = 0;
                cy = -1;
                while (cy == -1) {
                        i++;
                        if(i%2 == 0){
                                k = 0;
                                do{
                                        child = sh[ag][k];
                                        k++;
                                }while(ch1[child]==-1);
                                /*Cycle returns -1 if no cycle
                                    has been found and the
                                    position of the child if a
                                    cycle has been found*/
                                cy = cycle(v, child, i);
                                v[i] = child;
                                ag = child;
                        }

                        if(i%2 == 1){
                                k = 0;
                                do {
                                        school = ch[ag][k];
                                        k++;
                                }while(qs[school] == 0);
                                v[i] = school;
                                ag = school;
                        }
                }
                /*Solutioon returns how many seats have been
                    occuped*/
                cont -= solution(v, sol, ch1, qs, ns, nc, cy, i);
```

```
        }

/*Printing solutions*/
        printf("\n\nSOLUTION OF TOP TRADING CYCLE MECHANISM\n");
        printf("
            ------------------------------------------------------------\n");
        for(i=0; i<ns; i++){
                printf("SCHOOL%d = ", i);

                for(j=0; j<nc; j++){
                        if (sol[i][j] != -1)
                        printf(" %d ", sol[i][j]);
                }
                printf("\n");
        }

/*Free memory*/
        for(i=0; i<ns; i++){
                free(sh[i]);
                free(sol[i]);
        }
        for(i=0; i<nc; i++)
                free(ch[i]);
        free(v);
        free(qs);
        free(ch1);

        return 0;
}

int cycle (int* v, int child, int n){
        int i=0, cont=-1;
        while(v[i]!=child && i<n){
                i+=2;
        }
        if(i<n){
                cont = i;
        }
        return cont;
}

int solution (int *v, int **sol, int *ch1, int *qs, int ns, int
   nc, int beg, int end){
        int i, cont=0;
        for(i=beg; i<end; i+=2){
                ch1[v[i]]=-1;
                cont++;
```

```
                fillsol(sol, v[i], v[i+1]);
                qs[v[i+1]]-= 1;
        }
        return cont;
}

void fillsol (int **sol, int child, int school){
        int i = 0;
        while(sol[school][i]!=-1)
                i++;
        sol[school][i] = child;
        return;
}
```

Bibliography

[1] Abdulkadiroglu, A. and Sönmez, T. (2003) School Choice: A Mechanism Design Approach. *The American Economic Review* Vol. 93, No. 3, pp. 729-747.

[2] Abdulkadiroglu, A. and Sönmez, T. and Roth, Alvin E. and Pathak, Parag A. (2005) The Boston public school match. *The American Economic Review* Vol. 95, No. 2, pp. 368-371.

[3] Bayrak, E. and Sönmez, T. (2006) Games of School Choice under the Boston Mechanism. *Journal of Public Economics*

[4] Gale, D. and Shapley, Lloyd S. (1962) College admissions and the stability of marriage. *The American Mathematical Monthly* Vol. 69, No. 1.

[5] Gusfield, D. and Irving, R. W. (1989) *The stable marriage problem: structure and algorithms.* The Massachusetts Institute of Technology Press.

[6] Klijn, F. (2008) Mechanism design in school choice: some lessons in a nutshell. *Boletín de Estadística e Investigación operativa* Vol. 24, No. 3, pp. 11-22.

[7] Knuth, Donald E. (1981) *Stable marriage and its relation to other combinatorial problems.* Centre de Recherches Mathématiques de Montréal. Vol 10.

[8] Roth, Alvin E. (1982) The economics of matchings: stability and incentives. *Mathematics of operation research* Vol. 7, No. 4, pp. 617-628.

[9] Roth, Alvin E. (1984)The evolution of the labor market for medical interns and residents: a case study in game theory.*Journal of Political Economy* Vol. Y2, No. 6.

[10] Roth, Alvin E. (1985) The college admission problem is not equivalent to the marriage problem. *Journal of Economic Theory* Vol. 36, pp. 277-288.

[11] Roth, Alvin E. and Sotomayor, M. A. Oliveira *Two sided matching: A study in game-theoretic modelling and analysis.* Cambridge University Press, Cambridge 1992.

[12] Sönmez, Tayfun and Ünver, M. Utku*Matching, allocation and exchange of discrete resources.* Handbook of Social Economics, Vol. 1A, pp. 781-852, The Netherlands 2011.

[13] Shapley, Lloyd S. and Scarf H. (1974) On Cores and Indivisibility. *Journal of Mathematical Economics* Vol. 1, pp. 23-37.