Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας Τεχνολογιας Πληροφορικης και Υπολογιστων

Φιλαλήθεις Μηχανισμοί Ανταλλαγής Νεφρών

Διπλωματικη Εργασια

του

Ανδρέα Μάντη

Επιβλέπων: Δημήτρης Φωτάκης
Λέκτορας Ε.Μ.Π.

Εργαστηριο Λογικης και Επιστημης Υπολογισμων
Αθήνα, Μάρτιος 2013

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομεας Τεχνολογιας Πληροφορικης και Υπολογιστων
Εργαστήριο Λογικής και Επιστήμης Υπολογισμών

Φιλαλήθεις Μηχανισμοί Ανταλλαγής Νεφρών

Διπλωματική Εργασία

του

Ανδρέα Μάντη

Επιβλέπων: Δημήτρης Φωτάκης
Λέκτορας Ε.Μ.Π.

Εγκρίθηκε από την τριμελή επιτροπή την 26^{η} Μαρτίου, 2013

(*Υπογραφή*)	(*Υπογραφή*)	(*Υπογραφή*)
....	....	....
Δημήτρης Φωτάκης Λέκτορας	Ευστάθιος Ζάχος Καθηγητής	Αριστείδης Παγουρτζής Επικ. Καθηγητής

Εργαστήριο Λογικής και Επιστήμης των Υπολογισμών
Αθηνα, Μαρτιος 2013

(*Υπογραφή*)

. . . .

Ανδρεας Μαντης

Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών
Εργαστήριο Λογικής και Επιστήμης Υπολογισμών

Ευχαριστίες

Με την ευκαιρία της ολοκλήρωσης αυτής της διπλωματικής εργασίας, θα ήθελα να ευχαριστήσω τον κ. Ζάχο, Καθηγητή Ε.Μ.Π, κ. Παγουρτζή, Επίκουρο Καθηγητή Ε.Μ.Π και κ. Φωτάκη, Λέκτορα Ε.Μ.Π για τις γνώσεις και την σημαντική βοήθεια που μου προσέφεραν κατά τη διάρκεια των σπουδών μου.

Ιδιαίτερα θα ήθελα να ευχαριστήσω τον επιβλέποντα κ. Φωτάκη για τη συνεχή καθοδήγηση και ενθάρρυνση που μου έδωσε κατά τη διάρκεια της εκπόνησης της διπλωματικής αλλά και για τη σημαντική και θετική επιρροή που είχε στην απόφασή μου να επιλέξω τον κλάδο της θεωρητικής πληροφορικής.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου για την αγάπη και τη στήριξη που μου έδωσαν όλα αυτά τα χρόνια.

Περίληψη

Η μεταμόσχευση ενός υγιούς νεφρού είναι η καλύτερη γνωστή αντιμετώπιση σοβαρής ασθένιας νεφρών. Εφόσον οι άνθρωποι έχουν δύο νεφρά και μπορούν να επιβιώσουν με ένα, πολλοί ασθενείς έχουν κάποιο συγγενή ή φίλο που είναι διατεθιμένος να δωρίσει το νεφρό του. Παρόλα αυτά, δεν είναι όλοι δότες συμβατοί με τους ασθενείς. Αυτό μας δίνει την ευκαιρία ανταλλαγής νεφρών. Δύο ή περισσότερα ζεύγη ασθενή-δότη ανταλλάζουν νεφρά έτσι ώστε κάθε ασθενής να λάβει νεφρό από δότη άλλου ζεύγους. Στη διπλωματική αυτή χρησιμοποιούμε αυτή τη λύση στα πλαίσια του Αλγοριθμικού Σχεδιασμού Μηχανισμών για να μεγιστοποιήσουμε τον συνολικό αριθμό μεταμοσχεύσεων που μπορούν να γίνουν σε ένα σύστημα με πολλούς ασθενείς, εξάγοντας όμως πρώτα τις αληθινές πληροφορίες από τους συμμετέχοντες του συστήματος που πιθανόν να έχουν κίνητρο να δηλώσουν ψευδώς τα ιδιωτικά τους δεδομένα. Θα παρουσιαστούν διάφοροι αλγόριθμοι (μηχανισμοί) που εξαιτίας του σχεδιασμού τους θα δίνουν κίνητρα στους συμμετέχοντες να μας δίνουν τις αληθινές τους πληροφορίες πετυγχαίνοντας παράλληλα μια λύση όσο το δυνατόν πιο κοντά στη βέλτιστη. Πιο συγκεκριμένα θα εξεταστούν υπό αυτό το πρίσμα τα προβλήματα του Ταιριάσματος και του Καλύμματος Κύκλων.

Abstract

Transplantation of a healthy kidney is the best treatment today for severe kidney disease. Since humans normally have two kidneys and need only one to survive, many patients have a family member or friend willing to donate them a kidney. However, not all potential donors are compatible with their desired recipient. This raises the possibility of kidney exchange, in which two or more incompatible donor-patient pairs exchange kidneys such that each patient receives a compatible kidney from the donor of another patient. In this thesis, we use this solution in the context of Algorithmic Mechanism Design in order to maximize the total number of transplants in a large patient pool by first eliciting all the true information from participants who might have incentives to misrepresent their private data. We will present various algorithms (mechanisms) that are specifically designed to give incentives to the participants to tell the truth while simultaneously arriving at a nearly optimal solution. In this context we will examine more specifically variations of the problems of Matching and Cycle Cover.

Keywords

Kidney Exchange, Mechanism Design, Maximum Matching, Strategyproof, Truthful, Incentive compatible, Approximation, Randomization, Social choice.

Table of Contents

List of Figures

Chapter 1

Introduction

Kidney transplantation is the preferred treatment for serious kidney desease. Ideally, a patient will have a friend or family member who is willing to donate a kidney. Unfortunately, however, that friend is not always compatible with the patient. When that is the case, the potential donor is sent home and the patient normally gets on a cadaver waiting-list where it is uncertain whether or when he will get a kidney transplant. Kidney exchange programs are trying to solve this problem by registering in their database not only the patient but also his incompatible potential donor; we call such a couple an incompatible couple. If we find two incompatible pairs in our database such that each donor is compatible to donate a kidney to the patient of the other pair, then both donors are willing to donate a kidney to that patient, since that means their intended patient getting a kidney even indirectly. Therefore we can perform two transplantations that would not be possible otherwise. The system that aims to organize such exchanges in a large scale is called "Kidney Exchange". The main problem that Kidney exchange deals with, is to find a way to maximize the number of transplants using this solution.

The very first step, however, is to get the true information that we need to solve the problem. This issue arises because the participants of the system give us the information and have vested interests in the outcome of the solution; they may very well give us false information in order to serve their self-interests. We solve this problem in the context of "Mechanism Design". In "Mechanism Design" we design algorithms (or mechanisms) that elicit the true information from self-interested agents and try to solve the problem as efficiently as possible. A more detailed presentation of the kidney exchange problem will follow and then a brief introduction of "Mechanism Design".

1.1 The Kidney Exchange Problem

End Stage Renal Disease (ESRD) is a fatal disease unless treated with dialysis or kidney transplantation. Transplantation is the preferred treatment[1]. The current methods don't suffice for covering all the transplantation needs and the result is potentially unnecessary fatalities.

In the year 2012, there were more than 90,000 patients on the waiting list for cadaver kidneys in the United States of America. In 2011, 33,581 patients were added to the kidney waiting list, and 28,625 were removed from the list. Also, in the same year, there were only 11,043 transplants of cadaver kidneys and 5,771 transplants of kidneys of living donors in the U.S.A . Unfortunately, that year, 4,967 patients died while on the waiting list and 2,466 others were removed from the list as "too sick to transplant". As time passes, the numbers are worse; the situation is worse now than 10-20 years ago in the U.S.A.[2]

If there is one safe conclusion from the statistics just mentioned is that we need to have more effective and efficient methods for allocating kidneys, so as to cover the gap between the demand and the supply of kidneys. However, there is a wide-spread agreement that using money to obtain kidneys is unethical. Moreover, buying and selling organs is illegal in all countries except Iran. Therefore, a new way of allocating kidneys without using money would be extremely helpful.

First, we need to see how are the current practices of kidney transplantations. We know for a fact that transplants from live donors generally have a higher chance of survival than those from cadavers. The way such transplants are typically arranged is that a patient identifies a healthy willing donor (a spouse, for example) and two compatibility tests are carried out: the blood compatibility test and the tissue-type compatibility test. If both tests succeed and thus the transplant is feasible, the transplantation is curried out. If the transplant from the willing donor is not feasible, the patient typically enters (or remains on) the queue for a cadaver kidney, while the donor returns home.

A way to improve on this is called a paired exchange and it involves two patient-donor couples, for each of whom a transplant from donor to intended recipient is infeasible, but such that the patient in each couple could feasibly receive a transplant from the donor in the other couple. This pair of couples can then exchange donated kidneys. Compared with receiving cadaver kidneys at an unknown future time, this improves the welfare of the patients. In addition, it relieves the demand on the supply of cadaver kidneys, and thus potentially improves the welfare of those patients on the cadaver queue. Moreover, paired exchange can be extended to 3-way exchange by having the donor of the first couple to donate to the patient of the second couple, the donor of the second couple to donate to the patiend of the third couple and finally the donor of the third couple to donate to the patient of the first couple. In the same way it can be extended to 4-way exchange, 5-way exchange and so forth.

Another solution is called list exchange, where we have an exchange between one incompatible patient-donor couple, and the cadaver queue. In this kind of exchange, the patient in the couple receives high priority on the cadaver queue, in return for the donation of his donor's kidney to someone on the queue. This improves the welfare of the patient in the couple, compared with having a long wait for a suitable cadaver kidney, and it benefits the recipient of the live kidney, and others on the queue who benefit from the increase in kidney supply due to an additional living donor. However, Ross et al. note that this may have a negative impact on type O patients already on the cadaver queue, which makes the solution not generally acceptable.

According to Roth et al.[16] a crucial step in order to make these solutions have a substantial impact, is to make the Kidney Exchange Program a national system. We can understand why this is very important if we understand this problem as a market problem. Because of the money restriction we cannot use money as a medium of exchange in this market where we consider kidneys as "goods". Therefore, the Kidney Exchange Market is essentially a barter economy, where the participants directly exchange goods. Because of that we come face to face with William S. Jevon's classic problem of the "double coincidence of wants": "The first difficulty in barter is to find two persons whose disposable possessions mutually suit each other's wants. There may be many people wanting, and many possessing those things wanted; but to allow of an act of barter, there must be a double coincidence, which will rarely happen. The owner of a house may find it unsuitable, and may have his eye upon another house exactly fitted to his needs. But even if the owner of this second house wishes to part with it at all, it is exceedingly unlikely that he will exactly reciprocate the feelings of the first owner, and wish to barter houses. Sellers and purchasers can only be made to fit by the use of some commodity... which all are willing to receive for a time, so that what is obtained by sale in one case, may be used in purchase in another. This common commodity is called a medium, of exchange, because it forms a third or intermediate term in all acts of commerce." [3]

While the solution Jevons proposed was essentially money, we need to face this problem without using it. So, the solution is to make the market thick enough to alleviate the problem of double coincidents of wants. In simple terms, if we have enough people in the system, the problem of finding a suitable incompatible pair for exchange for someone would be almost non existent.

1.2 Mechanism Design

Mechanism Design is a subfield of economic theory that is interested in designing algorithms (or mechanisms) that incentivize the participants to give the true information that the mechanism needs for solving the problem, and at the same time achieve as efficient solution as possible. At first, we will introduce the goals of the designed mechanisms in the abstract terms of social choice. We will then show why such goals are impossible in the general case using the well known Gibbard-Satterthwaite Impossibility Result. Lastly, we will show how we escape this result in the case of Kidney Exchange. A more detailed analysis of the results of this section can be found in [4, 5, 9, 10, 47, 48].

1.2.1 Social Choice

Social Choice is the aggregation of individuals' preferences to a single joint decision. An example of that is elections. Many individuals ,called voters, have preferences over the candidates and at the end a single candidate must be elected. This is an easy problem to solve for 2 candidates: you just take the majority vote. But what happens for 3 candidates? We will show that the problem of social choice is not trivial and has some basic intrinsic difficulties in the general case. The first clue is Condorcet's paradox.

1.2.1.1 Condorcet's Paradox

Consider an election with 3 candidates: a, b and c. The election by majority vote is problematic. Here's why: Assume that we have 3 voters with the following preferences:

i $a \succ_1 b \succ_1 c$

ii $b \succ_2 c \succ_2 a$

iii $c \succ_3 a \succ_3 b$

For any candidate that is chosen, there is a majority of voters who prefer a certain candidate over the chosen one. That means that the majority of voters have incentives to lie about their preferences in order to change the outcome of the mechanism. That tells us that we cannot simply take a majority vote to aggregate the preferences of the voters.

1.2.2 The Gibbard-Satterthwaite Impossibility Result

In general, we have a set of alternatives A and a set of players N -what was previously the set of candidates and voters respectively. Let us denote L as the set of linear orders on A. Each player i has preferences $\succ_i \in L$, where $\succ_i$ a total order on A. A social choice function, which is implemented by a mechanism, maps the preferences to a single alternative. In Condorcet's Paradox, the mechanism used was just taking the majority vote. Note that all mechanisms are essentially algorithms. Although it was a trivial algorithm in this case, it is still an algorithm. Social choice functions and thus mechanisms must have some desired properties in order to be effective. More formally

Definition 1.2.1. *A function $f : L^n \to A$ is called a social choice function.*

We will now discuss some desired properties that mechanisms must have and then we will define them formally. One trivial but important property that a mechanism must have is to not rule out any alternative beforehand. For every alternative, there must be some set of the players' preferences that makes our mechanism return that alternative. Imagine how important that is. It could be the case that all the players prefer a certain alternative and that alternative is ruled out beforehand. Another important property is incentive compatibility (aka truthfulness, strategy-proofness). That means that every agent will not gain by misrepresenting his preferences to the mechanism. This is a crucial point. We need to get the best possible alternative and that will not happen if we do not have the true preferences of all the players in our hands. Lastly, the mechanism cannot let an agent force his top preference to the outcome of the mechanism, regardless of the other players' preferences; this unwanted property is called dictatorship. The formal definitions follow.

Definition 1.2.2. *Let f be a social choice function. Then if $\forall a \in A, \exists x \in L^n$ such that $f(x) = a$, we say that the social choice function is Onto.*

Definition 1.2.3. *A social choice function f can be strategically manipulated by player i if for some $\succ_1, ..., \succ_n \in L$ and some $\succ_i' \in L$ we have that $a \succ_i a'$ where $a = f(\succ_1$*

$, ..., \succ_i, ..., \succ_n)$ and $a' = f(\succ_1, ..., \succ'_i, ..., \succ_n)$. That is, voter i that prefers a' to a can ensure that a' gets socially chosen rather than a by strategically misrepresenting his preferences to be $\succ'_i$ rather than $\succ_i$. f is called incentive compatible (aka truthful or strategy-proof) if it cannot be manipulated.

Definition 1.2.4. *Voter i is a dictator in social choice function f if for all $\succ_1, ..., \succ_n \in L$, $\forall b \neq a$, $a \succ_i b \Rightarrow f(\succ_1, ..., \succ_n) = a$. f is called a dictatorship if some i is a dictator in it.*

What follows is very deep and negative theorem that forces us to change perspective.

Gibbard-Satterthwaite Theorem. *Let f be an incentive compatible social choice function onto A, where $|A| \geq 3$, then f is a dictatorship.*

Notice that the condition of Onto is weak. Actually, we don't just need a mechanism that is onto, we need a mechanism that is pareto-efficient. Pareto-efficiency means that the mechanism returns a maximal solution in a sense. If the mechanism is pareto efficient and returns an alternative there is no other alternative that everyone prefers instead of the one chosen. More intuitively, if you reach pareto efficiency, you cannot improve some player without hurting someone else. The formal definition follows. Notice that a Pareto Efficient mechanism is Onto but not the other way around.

Definition 1.2.5. *if $f(\succ_1, ..., \succ_n) = a$, then $\nexists b \in A$ such that $b \succ_i a, \forall i \in N$.*

The Pareto-Efficiency criterion is mainly used by economists to show that the best social outcome has beeen achieved. However, we can use another approach based on a Computer Science perspective. We use an objective function to model the social welfare and try to find an optimal solution; that is a solution whose value is the maximum value possible for the objective function. This criterion of social welfare optimization is stronger than pareto-optimality in the same manner that a maximum solution is stronger than a maximal solution. We will use both concepts throughout this thesis.

The main concern is to find ways to escape this impossibility result - even in its stronger versions of pareto-efficiency or social welfare optimization. In this thesis, we will attempt to review such a way for the problem of Kidney Exchange, considering a field that is an interface between Mechanism Design and Computer Science, called "Algorithmic Mechanism Design". This field was initiated by Nisan and Ronen [26] in 1999. A lot of work in this field is related to VCG Mechanisms [6, 7, 8] with which one can escape the impossiblity result. VCG Mechanisms use payments in order to compensate the participants when the outcome of the mechanism is not considered desirable for them. With this extra power in our hands, we can easily escape the impossibility result. Moreover, VCG mechanisms have two very desirable properties: social welfare optimization and truthfulness. However, these mechanisms have important drawbacks for which a large part of the research in this field is devoted to rectify using alternative models and techniques.

VCG mechanisms need to find the optimal solution of a problem, which in many cases is computationally intractable. A great amount of work has focused on designing truthful mechanisms that can be implemented efficiently, at a minimum loss of social efficiency, including combinatorial auctions [26, 27, 28, 29] and machine scheduling [30, 31, 32]. Another important drawback is that sometimes the use of VCG mechanisms may result in a huge amount of overpayment to the participants [30]. Therefore, a lot of work is focused on truthful mechanisms with small payments [30, 33, 34, 35, 36] or with a sharp budget constraint [37,38] or with no payments at all [39, 11, 40, 13, 41, 42].

How can we escape the impossibility result in the case of Kidney Exchange though? Since it is considered unethical and it is illegal in most countries to use money to solve the problem of Kidney Exchange, we cannot use any solution with payments. There are, however other changes we can do in the traditional model in order to escape the impossibility result. The impossibility theorem assumes that the players' preferences are completely unrestricted. That is not a realistic assumption under many settings. Facility-location [11, 40, 43, 49], is the problem where agents report their location in a metric space and the mechanism has to choose the location for a facility that is best for everyone. In this setting the player wants to be as close to a facility as possible. Therefore, we can safely assume, that he will not prefer a position of a facility to be farther instead of closer. Therefore we escape the impossibility result by essentially restricting the domain of preferences. In the case of Kidney Exchange, a patient cannot prefer a donor with whom is incompatible, thus restricting his domain of preference. Sometimes a patient's preferences are even modeled as a 0-1 preference. Other escape routes we will use is randomization and a slight relaxation of the strategy-proofness condition.

Another important tool we will use to achieve truthfulness in Kidney Exchange, is approximation. Procaccia and Tennenholtz [11] originated the subfield of "Approximate Mechanism Design Without Money" where one approximates socially optimal solutions in order to achieve strategy-proofness without using payments. Their setting was facility-location [11, 40, 43] which was mentioned in the previous paragraph. Other work in this field was done in job scheduling in the context of the Generalized Assignment Problem [35, 44, 45, 46]. For Kidney exchange in some settings, the restriction of the preferences is enough [15], but in the most interesting cases we will have to approximate the optimal solution in order to achieve strategy-proofness.

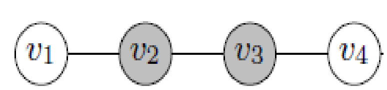
In the following section we will present the optimization problems that correspond with the Kidney Exchange problem without giving any attention to incentives, review some facts about their computational complexity and in the rest of this thesis we will view these problems in the light of Algorithmic Mechanism Design.

1.3 Optimization Problems and Computational Complexity

As we mentioned before, the problem of Kidney Exchange corresponds to some known computational problems studied in Computer Science. At the moment, the system of kidney exchange used in practice disregards incentives altogether and just solves the optimization problem in the traditional sense. Therefore, it is important to look at the computational complexity of these problems and how they are solved in practice. For an interesting look at the computational complexity of different variations of decision problems with incentives relating to Kidney exchange, see [50, 51, 52]

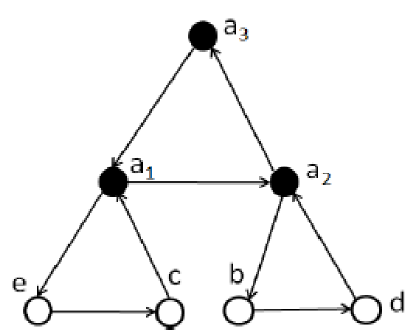
If we only allow pairwise exchanges, we fall into the optimization problem of Maximum Matching. The reason is that we can consider that the incompatible couples are the vertices of the graph and a pairwise exchange can be curried out only if there is an edge between the vertices. For example, in the figure below, there is a possibility for exchange between (v_1, v_2), (v_2, v_3) and (v_3, v_4) For this problem there is a known algorithm of Edmonds [53] which runs in polynomial time.

Figure 1.1: Example of pairwise exchange



If we allow more than 2-way exchanges, however, we have a problem. First, we have to change our model a bit. The edges are now directed and they indicate that the donor of the starting vertex can donate a kidney to the patient of the end vertex. So, a cycle of length k in this graph, is actually a k-way exchange. For example, in the figure below, there can be a 3-way exchange between (a_1, a_2, a_3), (a_1, e, c) and (a_1, b, d). So, the problem of Perfect Cycle Cover returns the exchanges that must be curried out in order to satisfy all the players. A more relevant version of cycle cover to the kidney exchange problem is the "short" cycle cover where we bound the length of the cycle. Unfortunately, this problem is proved to be NP-complete [12] using a reduction from the 3-D Matching problem. Moreover, if we allow weights on every edge and the aim is to maximize the total weight of cycles of length less or equal to 3, -which may simulate in practice the difference in priority between various patients- the problem is proved to be APX-complete [54]. That means that the best thing we can do is approximate it within a constant factor of the optimal solution in order to solve it in polynomial time. There exists a $(2 + \epsilon)$-approximation algorithm for this variation of the problem [54].

Figure 1.2: Example for k-way exchanges



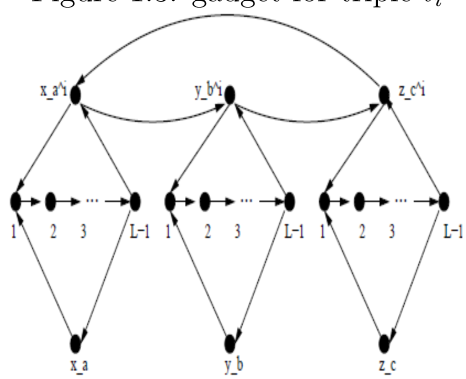
Theorem 1.3. *Given a graph $G = (V, E)$ and an integer $L \geq 3$, the problem of deciding if G admits a perfect cycle cover containing cycles of length at most L is*

NP-complete.

Proof. The problem is clearly in the NP. We can reduce the 3D-Matching problem to the perfect cycle cover containing cycles of length at most L (from now on called perfect short cycle cover). 3D-Matching is the problem that given disjoint sets X, Y, Z of size q, and a set of triples $T \subseteq X \times Y \times Z$, to decide if there is a disjoint subset M of T with size q.

Given an instance of 3D-Matching, construct one vertex for each element in X, Y and Z. For each triple, $t_i = \{x_a, y_b, z_c\}$ construct the gadget in Figure 1

Figure 1.3: gadget for triple t_i



Let M be a perfect 3D-Matching. If $t_i \in M$, then add the cycles of length L which include x_a, y_b, z_c and the cycle $< x_a^i, y_a^i, z_a^i >$ in the cover. If $t_i \notin M$, then add the cycles of length L which include x_a^i, y_a^i, z_a^i in the cover. Since M is a perfect matching all of the x_a, y_b, z_c are covered once and therefore we have a perfect short cycle cover since we only used cycles of length 3 or L.

Conversely, suppose we have a perfect short cycle cover. No short cycle can involve two distinct nodes from a different gadget. We can either use all the cycles of length L which include x_a, y_b, z_c or not at all; we cannot use only one or two such cycles; they are included or excluded together. Therefore, if we include such cycles of a specific gardet, x_a, y_b, z_c are not included on any other cycle, so they enter the disjoint subset with no problem. Otherwise, the cycles necessarily fall in the other case where $t_i \notin M$.

□

However, the medical infractructure does not allow as large exchanges as we want for incentive reasons. If we have a 2-way exchange, we must have 2 transplantations at the same time. What will happen if we don't? The first operation will take place and one of the two patients will have a kidney. Later, until the second operation starts, the second donor may change his mind, since his friend already has a kidney and thus the donor has no incentive to donate his anymore. Notice that 2 transplantations at the same time means 4 operations at the same time because we have nephrectomies before the transplantations. Imagine what will happen if we need to do that for a 10-way exchange. It is not feasible in practice. So, the medical community on this project agreed mainly on 2-way exchanges, although allowing 3-way exchanges is not too far from their capabilities. Moreover, allowing 3-way exchanges will have a significant impact on the total number of transplantations [16]. Therefore, the optimization problem of matching, as well as that of cycle cover with a bounded cycle length is crucial to the Kidney Exchange project.

Lastly, we will see how this optimization problem is practically implemented in most organizations that use kidney exchange. It as an Integer Linear Program with no regard to incentives, formulated as follows:

$$
\begin{aligned}
\text{maximize} \quad & \sum_{i \in N, j \in N} x_{i,j} \\
\text{subject to} \quad & x_{i,j} \in \{0,1\} && , \forall i, j \in N \\
& x_{i,j} \leq c^*_{i,j} && , \forall i, j \in N \\
& \sum_{j \in N} x_{i,j} \leq 1 && , \forall i \in N \\
& \sum_{j \in N} x_{i,j} = \sum_{j \in N} x_{j,i} && , \forall i \in N \\
& x_{i_1,i_2} + x_{i_2,i_3} + ... + x_{i_k,i_k+1} \leq k-1, && \forall \{i_1, i_2, ..., i_k, i_k+1\} \subset N
\end{aligned}
$$

where $x_{i,j} = 1$ means that patient i receives a kidney from donor j and $c^*_{i,j} = 1$ means that patient i is compatible with donor j. The second restriction suggests that one cannot receive a kidney if one is incompatible with the donor; the third restriction says the patient i can receive at most 1 kidney; the forth restriction suggests that the pair will give a kidney if and only if it receives one; lastly, the final restriction is about letting cycles of at most K length. The problem with this Integer program is not only that it is NP-complete but also that it needs too much memory. In instances of just 10,000 patients, using C-PLEX, memory becomes a problem before time. This is solved using realistic instances by Abraham et al. in [12] with a technique called "incremental problem formulation".

Chapter 2

Patients as Players of Kidney Exchange

In this chapter, we will examine the first attempt to model the Kidney Exchange problem in the context of Mechanism Design. When the Kidney Exchange Program was small and there was no central entity for collecting medical data, incentives of surgeons and patients were crucial. Surgeons could misrepresent the medical data of their patients in order to benefit their own patients, disregarding the general welfare of all the patients participating in the program. We will start off with the first model that was created by Roth et al. [14] and then we will present later models [15] which were refinements of the first one based on communications between economists and surgeons. On those models we will present effective mechanisms that were proposed for solving the problem of kidney exchange. One basic characteristic of these models -and all the later models in this thesis for that matter-is that they are static; the patient pool does not evolve over time. An interesting work on dynamic models has been done in [55, 56, 57]. Another characteristic is that we do not concern ourselves with justice in these models; we just want to maximize the social welfare. There is some work on egalitarian mechanisms in [14, 58] where the aim is to equalize the probabilites of the patients receiving a kidney. This is a useful approach when there is no exogenous way to distinguish among recepients.

2.1 General Model

Let $G(V, E)$ be the directed graph that represents the problem of Kidney Exchange. The set of nodes is $V = \{d_1, p_1, ..., d_n, p_n\}$, where $d_i s$ are the donors and $p_i s$ are their patients. For each i we know that d_i is a friend or family member of p_i but is incompatible with him; (d_i, p_i) is an incompatible pair with p_i being the intended patient of d_i. The set of directed edges is $E = \{(d_i, p_i)\} \bigcup \{(p_i, d_j)\} \bigcup \{(p_i, w)\}$, where the first set contains the edges from donors to their intended patients, the second set contains the edges from the patients to their prefered donor and the last set contains the edges from the patients to the waiting-list option. Thus, we have two kinds of edges: the edges that indicate the preferences of the patients and the edges that connect the icompatible donor-patient pairs.

Each patient has preferences over donors that must report to the system in order for the system to benefit the patients as best as it can. Formally, let P_i be the strict preference relation of p_i over all his compatible donors, d_i and the waiting-list option. For our purposes the relevant part of P_i is the ranking up to d_i or w whichever ranks higher. If d_i ranks higher, it means that if the patient does not get his prefered kidneys that rank above d_i, he prefers to wait for another exchange game to get a kidney, than to get the waiting-list option in this one. If w is higher, it means that if he doesn't get his prefered kidneys that rank above w, he prefers to get the waiting-list option, rather than wait for another exchange game. The option of d_i is not real because we already mentioned that d_i and p_i are incompatible. As we indicated above, it serves a different purpose than the rest of the $d_j s$

An m-way Kidney exchange is translated into a Cycle of length m in the graph. A *Cycle* is an ordered list of kidneys and patients $(d_1, p_1, d_2, p_2, ..., d_m, p_m)$ such that donor d_1 points to patient p_1 ,patient p_2 points to donor d_2, ... , donor d_m points to patient p_m, and patient p_m points to donor d_1 . Cycles larger than a single pair are associated with direct exchanges, very much like the paired-kidney-exchange programs, but may involve more than two pairs, so that patient p_1 is assigned kidney d_1 , patient p_2 is assigned kidney d_2 , ... , patient p_m is assigned kidney d_1 .Note that each kidney or patient can be part of at most one cycle and thus no two cycles intersect.

An indirect (list) exchange is translated into a so called w-chain. A w-chain is an ordered list of kidneys and patients $(d_1, p_1, d_2, p_2, ..., d_m, p_m)$ such that kidney d_1 points to patient p_1 , patient d_2 points to kidney p_2 , ... , kidney d_m points to patient p_m , and patient p_m points to w. We refer to the pair (d_m, p_m) whose patient receives a cadaver kidney in a w-chain as the head and the pair (d_1, p_1) whose donor donates to someone on the cadaver queue as the tail of the w-chain. W-chains are associated with indirect exchanges but unlike in a cycle, a kidney or a patient can be part of several w-chains. One practical possibility is choosing among w-chains with a well-defined chain selection rule, very much like the rules that establish priorities on the cadaveric waiting list.

The outcome of the mechanism will be a matching between donors and patients and between donors and the waiting list option. Please note that many patients can be matched to the waiting-list option whereas only one donor can be matched to a patient.

The criterion by which we judge the quality of our outcome in this model is Pareto Efficiency in regard to the incompatible pairs. That means given a kidney exchange problem, a matching is Pareto efficient if there is no other matching that is weakly preferred by all patients and donors and strictly preferred by at least one patient-donor pair. A kidney exchange mechanism is efficient if it always selects a Pareto efficient matching among the participants present at any given time.

One important assumption that we make in this model is that we can have as large exchanges as we want. That is an unrealistic assumption but this will be rectified in later models.

2.1.1 The TTCC Mechanism

We will introduce now the TTCC (Top Trading Cycle and Chains) Mechanism and then we will examine its properties. This mechanism is based on the TTC (Top Trading Cycle) Mechanism which applies to the problem of House Allocation first attributed to David Gale in [61] and later generalized by Abdulkadiroglou in [59]. The main ideas for the proofs of truthfulness and pareto-optimality are based on the work regarding the problem of House allocation with and witout existing tenants in [25, 59, 60, 61, 62].

Algorithm 1 TTCC Mechanism

---

1. Initially, all kidneys are available, and all agents are active. At each stage of the procedure each remaining active patient p_i points to his most preferred remaining unassigned kidney or to the wait-list option w, whichever is more preferred, each remaining passive patient continues to point to his assignment, and each remaining kidney d_i points to its paired recipient p_i.
2. If no cycles exist, go to step 3. In the case that cycles do exist, arrange the exchanges indicated by the cycles and erase all cycles and the participating donors and patients from the graph. Then, each remaining patient points to his top choice among the remaining active donors. Repeat step 2.
3. If we have no more nodes, we are done. Otherwise, we have only w-chains since we erased all the cycles in the previous step. Select only one chain according the well-defined selection rule and make the assignment of the participating patients final. Depending on the chain selection rule, we may choose to remove the participants or to keep the chain while making all the patients in the chain inactive. Then, each remaining patient points to his top choice among the remaining active donors. Cycles may form, so we repeat step 2.

The above mechanism works only if the graph has either cycles or w-chains, meaning that if we remove the cycles from the graph, there will be definitely at least one w-chain. In reality, if we remove all the cycles from the graph every node will be a part of a w-chain. The reason is simple. We know that all patients point to either w or a donor. Let us examine what happens if we start a path from an arbitrary node in the graph. We will never encounter a donor twice since there are no cycles. Therefore due to the finite number of nodes, the path will terminate at w.

Another thing that needs to be cleared concerns how we choose our w-chains. There are many ways to do this. For example:

a Choose minimal chains and remove them

b Choose the longest w-chain, and remove it. If the longest w-chain is not unique, then use a tiebreaker to choose among them.

c Choose the longest w-chain, and keep it. If the longest w-chain is not unique, then use a tiebreaker to choose among them.

d Prioritize patient-donor pairs in a single list. Choose the w-chain starting with the highest priority pair, and remove it.

e Prioritize patient-donor pairs in a single list. Choose the w-chain starting with the highest priority pair, and keep it.

f Prioritize the patient-donor pairs so that pairs with type O donor have higher priorities than those who do not. Choose the w-chain starting with the highest priority pair; remove it in case the pair has a type O donor, but keep it otherwise.

The last chain selection rule exists because the indirect exchange is considered unfair to O patients without incompatible pairs; this rule attempts to remedy this in some degree. Also, the reason why some rules choose to keep the chain in the graph is because one could extend the chain if need be; that is the donor that would offer his kidney to someone in the waiting list in a previous step of the mechanism, may offer it now to someone with an incompatible donor and **that** donor now offers **his** kidney to the waiting list instead of the previous one.

Theorem 2.1. *The TTCC Mechanism is Pareto Efficient and makes it a dominant strategy for patients to:*

- *Reveal their preferences over all available kidneys and the waiting-list option truthfully, and*
- *Declare their whole set of donors.*

That means that our mechanism meets all our desired properties. We will not attempt a formal and a complete proof, but rather we will describe the intuition behind the proof.

Firstly, we will discuss the reasons the TTCC Mechanism is Pareto-efficient. Any patient who leaves at step 1 is assigned her first choice and cannot be made better off. Any patient leaving at step 2 is assigned her best choice among those kidneys remaining and cannot be made better off without hurting someone who left at step 1. If we proceed in a similar fashion until the end of the mechanism, we conclude that no patient can be made better off without hurting someone who left at an earlier step. Thus, the TTCC Mechanism is Pareto-Efficient.

Secondly, we will examine the strategy-proofness of the mechanism. The second part is easy; if a patient does not declare his whole set of donors, she will have a smaller set of kidneys to set her preferences on. For the first part, assume that a patient leaves the mechanism at step k. Since she points to her best available kidney at each step, we know that all the kidneys that she prefers leave the mechanism before step k, and by misrepresenting, she cannot alter the cycles that have form at any step before step k. So, the better kidneys will leave before she does; at best she will hurt her own interests if she misrepresents her preferences. We can extend the argument (not in an obvious way) to include chains. Whether or not truthful preference revelation is a dominant strategy depends on the chain selection rule we use. We know it works for rule a, d, e and f.

2.2 Pairwise (2-way) Kidney Exchanges

The initial model was devised by Roth, Sonmez and Unver in 2004 [14] as a starting point to the project of creating a market for Kidney Exchange. In their subsequent discussions with medical colleagues, aimed at organizing such exchanges in the New England region of the transplant system, it became clear that a likely first step will be to implement pairwise exchanges [15], between just two patient-donor pairs, as these are logistically simpler than exchanges involving more than two pairs. That is because all transplantations in an exchange need to be carried out simultaneously, for incentive reasons, since otherwise a donor may withdraw her consent after her intended recipient receives a transplanted kidney. So even a pairwise exchange involves four simultaneous surgical teams, operating rooms, etc. Furthermore, the experience of American surgeons suggests to them that preferences over kidneys can be well approximated as 0-1, i.e. that patients and surgeons should be more or less indifferent among kidneys from healthy donors that are blood type and immunologically compatible with the patient. This is because, in the United States, transplants of compatible live kidneys have about equal graft survival probabilities, regardless of the closeness of tissue types between patient and donor. Furthermore, the list-exchange solution was discarted due to wide agreement that it was unfair to type O patients without an incompatible donor.

2.2.1 Pairwise Kidney Exchange Model

Let $N = \{1, 2, ..., n\}$ be the set of patients each of whom has one or more incompatible donors. Let $G(N, E)$ be the non-directed graph that represents the Kidney Exchange Problem. Each node of the graph represents a patient and its incompatible donor (an incompatible pair). Because we no longer assume that we have more than 2-way exchanges, the edges of the graph are non-directed and connect two nodes which are mutually compatible, that is the donor of the first pair can give his kidney to the patient of the second pair and the donor of the second pair can give his kidney to the patient of the first pair. Furthermore, since we have 0-1 preferences, an edge between two nodes exists if and only if those two nodes are mutually compatible.

The outcome of the desired mechanism will be a matching in the classic sense; a matching is a subset of the set of edges such that each patient can appear in at most one of the edges. For a matching M_μ we will use $\mu(a) = b$ to mean that we matched node a with b, and we will use $\mu(a) = a$ to mean that node a hasn't been matched.

The desired property of the matching that the mechanism must select is Pareto-Efficiency. In this context, a matching $\mu \in M$ is Pareto-Efficient if there exists no other matching $\eta \in M$ such that $M_\eta \supset M_\mu$.

The following results of abstract algebra help to understand the mechanism that follows and the reason why this mechanism has the desired properties. For notes on this, see [63]

Matroids and Matching

A matroid is a pair (X, I) such that X is a set and I is a collection of subsets of X (called the independent sets) such that:

M1 . if $A \in I$ and $B \subseteq A$ then $B \in I$; and

M2 . if $A \in I$ and $B \in I$ and $\|A\| > \|B\|$ then there exists an $i \in A \setminus B$ such that $B \cup \{i\} \in I$.

Proposition 1. *Let I be the sets of simultaneously matchable patients, i.e.* $I = \{I \subseteq N : \exists \mu \in M \text{ such that } I \subseteq M_\mu\}$. *Then* (N, I) *is a matroid.*

Lemma 1. *For any pair of Pareto-efficient matchings* $\mu, \eta \in M$, $\|\mu\| = \|\eta\|$

2.2.2 The Priority Mechanism

A Priority Mechanism produces a matching as follows for any priority ordering $(1, 2, ..., n)$ among the patients:

1 Let $E^0 = M$ (i.e the set of all matchings)

2 In general for $k \leq n$, let $E^k \subseteq E^{k-1}$ be such that:

$$E^k = \begin{cases} \{\mu \in E^{k-1} : \mu(k) \neq k\} & , \exists \mu \in E^{k-1} \text{ s.t } \mu(k) \neq k \\ E^{k-1} & , otherwise \end{cases}$$

For a given problem and priority ordering of the patients $(1, 2, ..., n)$ we refer to each matching in E^n as a Priority Matching, and a Priority Mechanism is a function that selects a Priority Matching for each problem.

In other words, we may describe the priority mechanism as follows:

Algorithm 2 Priority Mechanism

1: Given a priority ordering of the patients and at first $i = 1$, match priority i patient if she is mutually compatible with a patient in addition to all the previously matched patients; skip her otherwise.
2: Repeat step 1 for the patient with the priority $i + 1$ unless $i = n$

We conclude this section proving that the Priority Mechanism has all the desired properties regarding efficiency and incentives. The latter is proved using matroids and the former is derived immediately from the mechanism's construction.

Theorem 2.2. *A Priority Mechanism is Pareto Efficient, utility-maximizing and makes it a dominant strategy for patients to:*

- *Reveal their whole set of compatible kidneys, and*
- *Declare their whole set of donors.*

Proof. Let us examine without loss of generality the patient with the k-th priority. Let R be the kidney exchange game where the patient reveals their whole set of compatible kidneys. If under our mechanism ϕ the patient was matched, then she would have nothing to gain by reporting only a subset of her compatible kidneys. Otherwise, if the patient was not matched, let Q be the same game with the difference that the patient declares some of her compatible kidneys to be incompatible. Observe that this implies $E^{k-1}(Q) \subseteq E^{k-1}(R)$. Let $\phi(Q) = v$. Since $\mu(k) = k$,$\mu'(k) = k$ for all $\mu' \in E^{k\text{bffl}''1}(R)$. But then $\mu'(k) = k$ for all $\mu' \in E^{k\text{bffl}''1}(Q)$ as well and hence $v(k) = k$. Therefore, it is a dominant strategy for her to reveal her whole set of compatible kidneys.

What about declaring her whole set of donors? Well, according to the proof above a patient never suffers from enlarging her set of mutually compatible kidneys. Therefore, since declaring more donors will enlarge her set of mutually compatible kidneys, it is a dominant strategy for her.

Lastly, the Priority Mechanism is Pareto-Efficient from construction.A priority matching matches as many patients as possible starting with the patient with the highest priority and following the priority ordering, never "sacrificing" a higher priority patient because of a lower priority patient. Proposition 1 implies, through the second property of matroids, that the "opportunity cost" of matching a higher priority patient will never be more than one lower priority patient who could otherwise have been matched. And by Lemma 1, the same total number of patients will be matched at each Pareto-efficient matching, so there is no trade-off between priority allocation and the number of transplants that can be arranged. So, the Priority Mechanism is not just Pareto-efficient, but it also maximizes the utility of all the agents, that is the total number of transplants.

□

Chapter 3

Hospitals As Players of Kidney Exchange

As kidney exchange games begin to grow, we see that the incentives of the patients/-donors/surgeons do not play an essential role in the outcome of the game. The reason is that tests of tissue-type compatibility are standardized and thus the patients or their surgeons cannot manipulate the set of the compatible donors. Rather, the hospitals or the transplant centers have incentives to submit false information to the central matching mechanism [20]; they can match for example some pairs internally and reveal only the hard-to-match pairs to the mechanism. This behaviour has been already observed. So, from this point forward we will only consider incentives of hospitals.

3.1 The Model

We will use the term compatibility graph to describe the directed graph that has the incompatible pairs as nodes (meaning the patient and its incompatible donor) and edges (a, b) if and only if the donor of the incompatible pair a is compatible to donate a kidney to the patient of the incompatible pair b.

An exchange is a cycle in the graph so that every node in the cycle gets and gives a kidney. A matching M is called k-efficient if it matches the maximum number of transplants possible for exchanges of size no more than k, and k-maximal if there in no other matching M' such that $M' \supset M$. The terms maximal and efficient are defined the same way as above with the sole difference the k is no longer bounded.

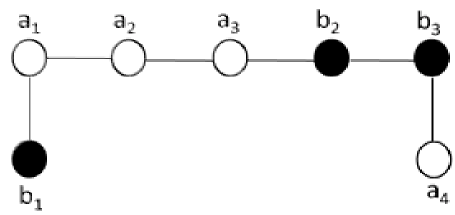
The basic difference with previous models is that we consider the incentives of hospitals rather the incentives of patients and thus the set $N = \{1, ..., n\}$ is now the set of hospitals participating in the exchange and each one controls a subset of nodes. Therefore, we use the notion of individual rationality to mean that the mechanism matches at least as much patients for each hospital as that hospital would have matched internally on its own. A more powerful notion is that of strategyproofness (or, truthfulness) which induces hospitals to prefer to reveal all their incompatible pairs rather than to reveal only some of their pairs to the mechanism. Notice that the second notion is more powerful because, in the first case we only demand that the strategy of participate fully be better than the strategy of not participating at all, and in the second case we demand that the strategy of participating fully be better than the strategy of not participating at all and also of the strategies of partially participating (thus, better than all the possible strategies).

3.2 Lower Bounds

Firstly, we need to show that approximation is a necessary evil for all mechanisms. One cannot have a mechanism that is both maximal and strategy-proof [20].

Proposition 3.1. *No Individually Rational mechanism is both maximal and strategy-proof.*

Figure 3.1: Graph used in proofs



Proof. Let us have two hospitals a and b such that $V_a = \{a_1, a_2, a_3, a_4\}$ and $V_b = \{b_1, b_2, b_3\}$ are the hospital a's nodes and hospitals b's nodes respectively.Note that every maximal allocation leaves exactly one node unmatched. Suppose ϕ is both maximal and IR. We show that if a and b submit V_a and V_b respectively, at least one hospital strictly benefits from withholding a subset of its nodes. If the node that is left unmatched belongs to hospital a, that means that hospital a will have a utility of 3. Hospital a however can hide a_1 and a_2 and then the maximal mechanism will match a_3 to b_2 and a_4 to b_3 because if it didn't it wouldn't be maximal. In that case the utility of hospital a would be 4 instead of 3. Therefore, it is in hospital a's best interest to hide some of the nodes and thus making the maximal mechanism not strategy-proof. A similar argument holds in the case that hospital b is the one with the node unmatched. $\square$

Note that if a mechanism is efficient,it is also maximal; so the result above also applies to efficient mechanisms as well. Another thing to observe is that, as we mentioned above, any strategy-proof mechanism is also individually rational, so we could leave the individual rationality out of the formulation of the proposition. Now we know that we must resort to approximation, the question is how much. Using basically the same instance of the problem as above, we can prove some lower bounds on the approximation ratio differentiating between deterministic and randomized mechanisms as shown in the theorem that follows [20, 13].

Theorem 3.1. *If there are at least two agents,*

1 . no deterministic strategy-proof mechanism can provide an α-approximation with respect to social welfare for $\alpha < 2$, and

2 . no randomized strategy-proof (in expectation) mechanism can provide an α-approximation with respect to social welfare for $\alpha < 8/7$.

Proof. 1.We will also use figure 1 for the proof of this theorem. As we mentioned above an efficient mechanism will have the following possible outcomes: either $G_a \leq 3$, or $G_b \leq 2$ where G_a is the gain of hospital a and G_b is the gain of hospital b. Let the mechanism to have approximation $\alpha < 2$. If hospital a decides to hide a_1 and a_2 as above, the mechanism having only the choice to match 2 or 4 nodes, matches 4 nodes because of the approximation assumption. Therefore, the total gain of hospital a will be 4 and the mechanism is not truthful. We can use a similar argument for the second case.

2.As far as randomized mechansims are concerned we can cover all the possible mechanisms with two cases: 1) a vertex of a is not matched with probability greater than 1/2 and 2) a vertex of b is not matched with probability greater than 1/2. We can see that we covered every mechanism if we observe that if case 1 doesn't hold, we necessarily fall in the second case. The expected gains in each case can be worked out to be: 1) $\mathbb{E}(G_a) \leq 3.5$ and 2) $\mathbb{E}(G_a) \leq 2.5$. Let us examine the first case where hospital a has at least one vertex uncovered with probability greater than 1/2. The strategyproof mechanism will match both of a's pairs with probability at most 3/4, for a maximum of 7/4 pairs in expectation, while the optimum is 2. Therefore, we have a 8/7-approximation mechanism. A similar argument holds in the second case. □

Note that although the specific instance of the problem we used in our proofs has only 2-way exchanges, the same result applies to larger than 2-way exchanges. Also, in the case of the randomized mechanism, we used the notion of truthfulness in expectation. We could use a much stronger notion: universal truthfulness. A mechanism is universally truthful (or universally strategyproof) if whatever the random choice of the mechanism, it is still the best stategy to be truthful even if the player knows the random choice. As we will see from the following theorem [22] if we demand that our mechanism is universally truthful we have worse approximation ratios in the case of 2 agents.

Theorem 3.2. *Let A be a randomized mechanism for 2-agent kidney exchange.*

1 . If A is universally truthful, then its approximation ratio is at least 3/2*.*

2 . If A is universally truthful and inclusion-maximal, then its approximation ratio is at least 2*.*

3 . if A is truthful in expectation and inclusion-maximal, then its approximation is at least 4/3*.*

What happens to our approximation bounds if we weaken the notion of truthfulness and demand only individual rationality? If we restrict our attention to 2-way exchanges, we see that there exist 2-efficient mechanisms [20]. The reason for that is that in 2-way exchanges, we are dealing with the problem of "classical" Matching, where a maximal allocation due to the properties of matroids is also a maximum allocation. The idea is that we first match the maximum number of vertices inside and for, each hospital and then we try to maximize the number of edges without unmatching previously matched vertices. That is done with the augmenting algorithm of Edmonds [53] in polynomial time.

Theorem 3.3. *There exists an individually rational allocation with exchanges of size at most 2 that is also 2-efficient in every compatibility graph.*

If we allow more than 2-way exchanges, the basic structure of the problem changes and we have a very bad approximation ratio. We no longer have the structure of a matroid because k-way, with k>2, exchanges are directed cycles of length k and not just an undirected edge as was the case with 2-way exchanges. The idea is that for almost each vertex in a k-cycle of a maximal allocation, we may have a large exchange that does not belong to the maximal allocation but to the efficient allocation. This possibility results in a very bad approximation ratio in the worst case [20].

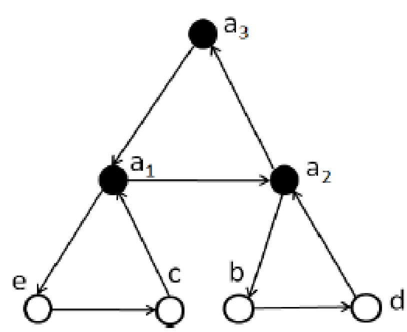
Theorem 3.4. *For every $k \geq 3$, there exists a compatibility graph such that no k-maximal allocation which is also individually rational matches more than $1/k-1$ of the number of nodes matched by a k-efficient allocation. Furthermore in every compatibility graph the size of a k-maximal allocation is at least $1/k-1$ times the size of a k-efficient allocation.*

Proof. Let M be a k-efficient allocation and M' an individually rational allocation. Since M' is individually rational, we can assume that it is also k-maximal, because we can match at first the maximum number of vertices for each hospital and then try to maximize the total number of vertices under that restriction without unmatching previously matched ones.

We know that every exchange in M must intersect with an exchange in M', because if there is an exchange in M that doesn't, then M' could use that exchange to increase the total number of vertices matched without affecting any of the other exchanges. But then, M' wouldn't be k-maximal.

So, let us fix an exchange in M' of size l with $2 \leq l \leq k$. We need to construct the way that M could have the largest possible difference from M'. That can happen if M has exchanges of the largest size possible (namely, k) that intersect each vertex in the fixed exchange. But then, M' could use the same allocation as M and increase its cardinality, contradicting its maximality again. Well, if we just remove one such exchange of M, leaving only one node of the fixed exchange of M' to not be intersecting with an exchange of M, M' can no longer increase its cardinality. Therefore, M matches $(l-1)k$ nodes where M' matches l nodes. So the ratio is $\frac{l}{(l-1)k}$, where in the best case $l = k$, the ratio becomes $\frac{k}{(k-1)k} = \frac{1}{(k-1)}$ which is the desired result.

Figure 3.2: Tight case for k=3



To see that the result is tight, we only need to see that a graph could consist only of the construction above. We can see that clearly for $k = 3$ in the figure.

□

Chapter 4

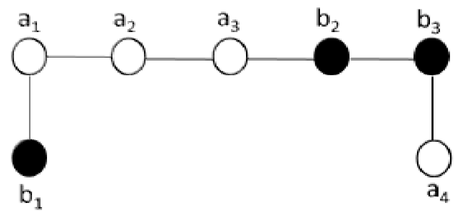
Deterministic Mechanisms: $MATCH_{\Pi}$

As proved in the previous chapter the lower bound of approximation ratio for deterministic strategy-proof mechanisms is 2. So, we need to design a mechanism that is both strategy-proof and as close to that ratio as possible. We will present the deterministic $MATCH_{\Pi}$ mechanism which is analyzed in [13].

We already know that strategy-proof mechanisms are individually rational, but since individual rationality is a weaker concept, not all individually rational mechanisms are strategy-proof. However, what is important is that in both cases we need to ensure that we can match as many patients for each hospital, as that hospital would have matched on its own. With that in mind, we come across the first important idea for a strategy-proof mechanism: Above all, include in the final matching, the maximum number of internal edges; that is the edges between two nodes of the same player.

Let us examine the mechanism that uses this idea and has the maximum possible cardinality. This mechanism considers at first all the matchings that maximize the number of internal edges; of those matchings, he considers those with the maximum overall cardinality; and finally he breaks ties serially and outputs a single matching. Is this "natural" mechanism strategy-proof? Let us examine at first what happens for two players in the example of the previous chapter.

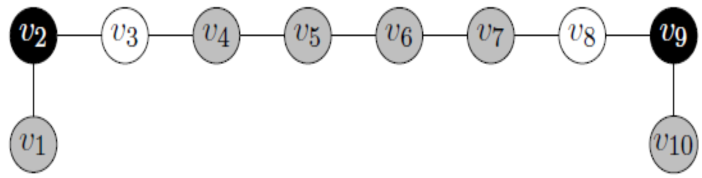
Figure 4.1: Graph used in proofs



A matching that maximizes the number of internal edges of agent 2 must include (b_2, b_3) and a matching that does the same for agent 1 must include either (a_1, a_2), or (a_2, a_3). Therefore all the matchings that maximize internal edges are $M_1 = \{(b_2, b_3), (as_1, a_2)\}$ and $M_2 = \{(b_2, b_3), (a_2, a_3), (a_1, b_1)\}$, from which M_2 has maximum cardinality and thus is finally chosen by the mechanism. In the previous chapter, agent 1 benefits if he chooses to hide (a_1, a_2). In this case, however, he cannot benefit because our mechanism having as input the graph where a_1 and a_2 are not visible, returns $M' = \{(b_2, b_3)\}$. Therefore, the gain of agent 1 is now 2, but if he were to tell the truth, his gain would be 3 as we have seen above. This example may seem to show that we probably have a strategy-proof mechanism in our hands, but as we will see in the next example, that is not the case.

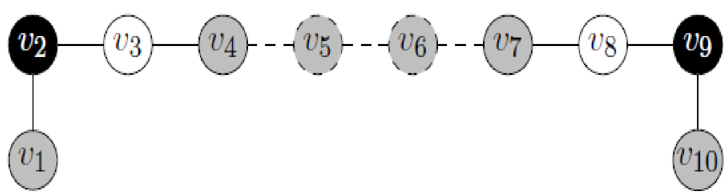
Let's examine figure 5 where agent 1 has the white vertices, agent 2 has the gray vertices and agent 3 has the black vertices. Agent 2 is the one who will lie. Any

Figure 4.2: Counter-example real graph



matching that maximizes the internal edges of agent 2 must include (v_4, v_5) and (v_6, v_7). Obviously, as above, it suffices to examine only the maximal matchings that contain these internal edges (because only maximal matchings have the chance to be maximum under these restrictions), that is $M_1 = \{(v_1, v_2), (v_4, v_5), (v_6, v_7), (v_9, v_{10})\}$, $M_2 = \{(v_2, v_3), (v_4, v_5), (v_6, v_7), (v_8, v_9)\}$, $M_3 = \{(v_1, v_2), (v_4, v_5), (v_6, v_7), (v_8, v_9)\}$ and $M_4 = \{(v_2, v_3), (v_4, v_5), (v_6, v_7), (v_9, v_{10})\}$. All of these matchings have the same overall cardinality, so we need to break ties to isolate the one that our mechanism will output. We break ties serially, so we give priority to agent 1 above all others. Therefore, we finally get M_2. Agent 2 has a gain of 4 vertices.

Figure 4.3: Counter-example graph when agent lies



Now, let us assume (Figure 6) that agent 2 lies and hides (v_5, v_6). Since we have no internal edges, we get the maximum matching: $M' = \{(v_1, v_2), (v_3, v_4), (v_7, v_8), (v_9, v_{10})\}$. In this case, agent 2 has a gain of 6 vertices. Therefore, our mechanism is not strategy-proof.

It is clear that since we maximize the number of internal edges of each agent, the problem lies in the matching of external edges of an agent. In this example, the agent by manipulating, he gets 4 extra external edges, namely (v_1, v_2), (v_3, v_4), (v_7, v_8) and (v_9, v_{10}). If we could adequately limit the extra external edges that the agent is able to induce the mechanism to match by lying, we would solve our problem. We must somehow balance the number of an agent's external edges that the mechansism matches in the graph induced by lying with the number of external edges that the mechanism matches in the "true" graph. The next mechanism achieves just that by using a bipartition of all the agents $\Pi = (\Pi_1, \Pi_2)$ and limiting the external edges only between agents belonging to different sides of the bipartition.

4.1 $MATCH_{\Pi}$

Algorithm 3 $MATCH_{\Pi}$

1: Given a graph G, consider all the matchings that have maximum cardinality on each V_i and do not have any edges between V_i and V_j when $i, j \in \Pi_l$ for some $l \in \{1, 2\}$, i.e., those that maximize the number of internal edges and do not have any edges between sets on the same side of the bipartition

2: Among these matchings select one of maximum cardinality, breaking ties serially in favor of agents in Π_1 and then agents in Π_2.

In order to examine strategyproofness, we need to consider the symmetric difference of the matching given by our mechanism when the "true" graph is the input and the matching given by our mechanism when an agent lies, namely $M \Delta M' = (M \cup M') \backslash (M \cap M')$. The reason for that is twofold: firstly, we only care about the difference of the two matchings and secondly, the symmetric difference has a nice structure that aids our proofs. The next lemma is very useful and it depicts the helpful structure of the symmetric difference of two matchings.

Lemma 4.1. *Let G be a non-directed graph and M_1,M_2 two matchings on G. Then, the graph*

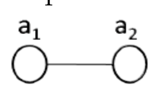
$$M_1 \Delta M_2 = (M_1 \cup M_2) \backslash (M_1 \cap M_2)$$

consists of vertex-disjoint paths (some of which maybe cycles) with alternating edges of M_1 and M_2.

Proof. We will use induction on the number of edges. Just as a reminder, given a graph $G = (V, E)$ a matching $M \subseteq E$ on G is a subset of edges such that each vertex is incident to at most one edge of M.

$m = 1$

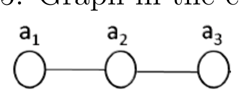
Figure 4.4: Graph in the case m=1



In this case, we just have one edge between two vertices. If both matchings get the edge (a_1, a_2), then $M_1 \Delta M_2$ would contain the empty graph. If $M_1 = \{(a_1, a_2)\}$ and $M_2 = \emptyset$ then $M_1 \Delta M_2 = \{(a_1, a_2)\}$ which is a path of length 1. The result holds for $m = 1$ almost vacuously.

$m = 2$

Figure 4.5: Graph in the case m=2



We will ignore the two identical matchings again because in that case $M_1 \Delta M_2 = \emptyset$. If $M_1 = \{(a_1, a_2)\}$ and $M_2 = \{(a_2, a_3)\}$, then $M_1 \Delta M_2 = \{(a_1, a_2), (a_2, a_3)\}$, which is a single path of length 2 with alternating edges ((a_1, a_2) belongs to M_1 and (a_2, a_3) belongs to M_2).

Let's assume that the result holds for $m-1$ edges.

Let's add a new edge to the graph $e=(u,v)$. If both matchings use it, it wouldn't add anything to $M_1 \Delta M_2$. So, let's say that M_1 uses it and M_2 doesn't. We can assume of course that in that case, M_1 does not match u or v any other way, because if it did, it would have one vertex incident to two edges which contradicts the definition of matching. We will examine three cases: either M_2 does not match any of u and v, or it matches both u and v, or it matches one of u and v.

If M_2 matches non of the u and v, then we would add to $M_1 \Delta M_2$ the edge e which is of course a path of length 1 with alternating edges (vacuously true) which does not intersect with any of the other paths assumed in the inductive hypothesis.

Let's assume that M_2 matches u and not v without loss of generality. Any path that contains e, cannot be a cycle and e must be one of its extreme edges, because of the fact that v is matched only by M_1 through e and not by M_2. Since ,by the inductive hypothesis, all the paths before the addition of e are vertex-disjoint, that means that only one such path contains u. That path also has alternating edges and an edge of M_2 matching u. Since we add e that is matched by M_1, the new, extended path has still alternating edges and does not intersect any other path in $M_1 \Delta M_2$.

Let's finally assume that M_2 matches both u and v. We know that these two vertices belong to different paths in $M_1 \Delta M_2$ before the addition of e. Then, we could either create a single cycle or a path that has alternating edges and does not intersect with any other path in $M_1 \Delta M_2$ by merging the two previously disjoint paths in a manner similar to the previous case.

$\square$

For instance, in the counter-example of our first attempt to define a strategy-proof mechanism (showing in Figure 5 and 6), $M_2 \Delta M'$ contains (v_1, v_2) which belongs to M', (v_2, v_3) of M_2, (v_3, v_4) of M' and (v_4, v_5) of M_2, which is an alternating path from v_1 to v_5. In the same way, an alernating path disjoint from the first one, starting from v_6 and ending on v_9 also belongs in $M_2 \Delta M'$. With this lemma in our hands, we can now proceed to the proof of the strategy-proofness of the $MATCH_\Pi$ mechanism.

Theorem 4.2. *For any number of agents, and for any bipartition* Π *of the set of agents,* $MATCH_\Pi$ *is strategyproof.*

Proof. Fix a bipartition of the agents Π and let G be the graph that includes all the patients of each hospital and G' is the graph that is induced by G if agent i hides some nodes . We define M as the matching that $MATCH_\Pi$ outputs on the graph G and M' as the matching that $MATCH_\Pi$ outputs on graph G' plus the edges that agent i used to match his hidden vertices internally.

We will consider the graph $M \Delta M'$ which, as proved in *Lemma* 11.1, contains vertex-disjoint paths or cycles with alternating edges of M and M'. Without loss of generality, we will consider one such path, since if we prove that the gain of agent i in M is at least as much as the gain of i in M' in this one path, it would mean that the result would hold in each vertex-disjoint path seperately. That means that the total gain of i is better in M rather than M' since all the edges outside $M \Delta M'$ are common to both matchings.

We can assume that our path is not a cycle without loss of generality. Assume, that the path is a cycle of odd length. The edges in the cycle are alternating. Therefore we will necessarily have one vertex to which two different edges of the same matching are incident, which contradicts the definition of matching. Now, assume that the path is a cycle of even length. That means that both matchings match all the vertices in the cycle, which makes us indifferent of this cycle in our attempt to find the difference of gain between the two matchings.

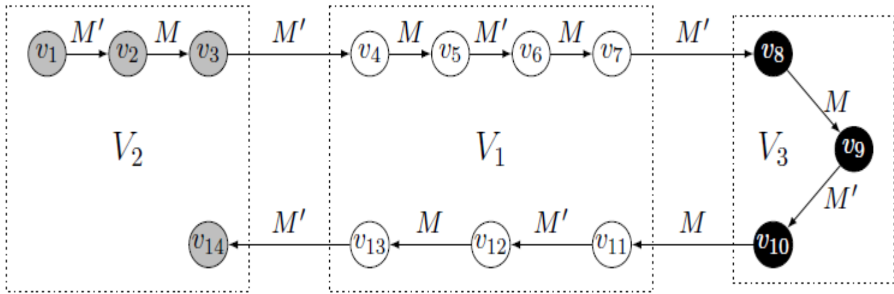
$MATCH_{\Pi}$ maximizes the internal edges and therefore we have 2 cases: Either that M matches more internal edges of i than M', or the same number of edges.From now on, we may use the term "enters" or "exits" in a subpath and so on, giving the wrong impression of a directed graph. That is not the case; those terms are used for simplicity of expression. We will use the symbol M_{ij} to denote the subset of matching M that contains edges between a vertex of agent i and a vertex of agent j.

1. $|M_{ii}| > |M'_{ii}|$

   Both M and M' maximize the number of internal edges of agents other than i. That means that every subpath of the path $M \Delta M'$ that enters and exits V_j, where $j \neq i$, has an even length. Assume that it didn't. That means for some matching, either M or M' there are more internal edges that belong to that matching than the other on that subpath. That means that the other matching could switch its edges with those of the better matching and increase its cardinality. That, however, contradicts the fact that both of these matchings are maximum inside each agent. Obviously, since these subpaths are even, that means that if an edge of M enters that subpath, an edge of M' necessarily exits it, or if an edge of M' enters that subpath, an edge of M exits it.

   As shown in the figure, agent 1 is the one who lies and $M_{ii} = 3$, whereas $|M'_{ii}| = 2$. That means that the graph of the figure represents the $M \Delta M'$ in the case we are considering. Also, note, as we mentioned before, that the arrows

Figure 4.6: $|M_{ii}| > |M'_{ii}|$



in the edges do not indicate a directed graph; they just exist for simplicity of expression. As expected, the subpath of V_3 has an even number of edges and also an edge of M' enters the subpath while an edge of M exits the subpath.

What follows is the essential difference between $MATCH_{\Pi}$ and the previous attempt for a strategyproof mechanism. We claim that if an edge of M' exits V_i, then an edge of M enters it again and if an edge of M exits V_i, then an edge of M' enters it again. That is easy to see in figure 9, since an edge of M' exits V_1 and an edge of M enters it later. Note that we cannot be sure what is going on a subpath of V_i in constrast with subpaths of V_j, $j \neq i$, because of agent i lying. For example, in figure 9, one subpath of V_1 has even length and the other subpath has odd length.

We will now proceed to prove our claim. Let us say, without loss of generality that agent i belongs to Π_1 and an edge of M exits V_i. That edge must enter a subpath of an agent V_j belonging to Π_2. We know that if an edge of M enters V_j, $j \neq i$, an edge of M' must exit V_j. So, an edge of M' exits V_j. That edge must enter a subpath of V_k belonging to Π_1 again. If that happens to be agent i, we proved our claim; if not, we will repeat the process described above, always having an edge of M' entering subpaths beloning to agents of Π_1. Since agent i belongs to Π_1, the path will re-enter V_i (if it does) with an edge of M'.

We now see why the presence of the bipartition limitation makes our mechanism strategyproof. It balances the number of external edges belonging to M and M'. The only external edges of V_i we haven't accounted for yet, are the first and last external edge of V_i, meaning the edge that entered V_i without ever entering before and the edge that exits V_i without ever entering again. As we see in figure 9, those edges are (v_3, v_4) and (v_{13}, v_{14}) and they belong both in M'. However, that is not necessary; in the general case they may both belong to M, or the one edge in M and the other in M', or at least one of them might not exist at all if we start or end at V_i. Therefore, it holds that,

$$\sum_{j \in N \setminus \{i\}} |M_{ij}| \geq \sum_{j \in N \setminus \{i\}} |M'_{ij}| - 2$$

Using that result, we finally prove that

$$u_i(M) = 2|M_{ii}| + \sum_{j \in N \setminus \{i\}} |M_{ij}| \geq 2(|M_{ii}| + 1) + (\sum_{j \in N \setminus \{i\}} |M'_{ij}| - 2) = u_i(M')$$

2. $|M_{ii}| = |M'_{ii}|$

Since M maximizes internal edges, in this case, M' also maximizes internal edges. As we have seen before, all the subpaths of V_j in which both M and M' maximize their internal edges are of even length. Using the same argument in this case, if a subpath of V_i has odd number of edges, then it has more edges of

M (M') than of $M'(M)$, then $M(M')$ could have switched its edges for those of $M'(M)$ and increase its cardinality, but since $M(M')$ maximizes the internal edges of agent i, that is not possible and therefore the subpath of V_i is of even length. Thus, all the subpaths of all V_j are of even length in this case.

Since both M and M' maximize internal edges and M is the maximum-cardinality matching under the constraint of maximum number of internal edges, $|M| \geq |M'|$. Let's consider firstly the case of $|M| > |M'|$. Our path in this case is of odd length and it starts and ends with an edge of M, otherwise we wouldn't have $|M| > |M'|$. We also know, that every subpath of V_i has odd length. That means that if we enter the subpath with an edge of M, we will exit with an edge of M' and if we enter with an edge of M', we will exit with an edge of M. Therefore, if all subpaths of V_i are not at the start or end of the path, both M and M' will match the same number of vertices of agent i. What happens if we start with a subpath of V_i? Since, we necessarily start with an edge of M, and exit the first subpath with an edge of M, M will match more vertices than M'. Now, if we end with a subpath of V_i, then we would necessarily end with an edge of M and enter V_i for the last time with an edge of M and therefore M would match more vertices of agent i than M'.

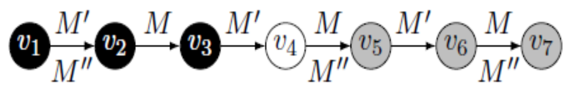
We can now assume that $|M| = |M'|$. That means that our path is of even length. That means that both matchings match all the vertices except the first and the last one. The only way agent i could manipulate the mechanism and by matching the first vertex with M' and the last vertex of the path would belong to another agent. The other vertex must necessarily belong to another agent, because M matches that vertex (because of the even length), and if it belonged to i, then his gain on M and M' would be identical. So we have a path of even length that start with a vertex of agent i and an edge of M' and ends with a vertex of agent j, $j \neq i$ with an edge of M. We will show, using proof by contradiction that such a path can never exist as a result of our mechanism, and therefore agent i cannot gain by lying.

Let's assume that the mechanism breaks ties in favor of i over j. M and M' have the same cardinality overall and maximize all internal edges. Their only difference is that M matches one more vertice of j and one less vertices of i, than M'. If M would switch to M', then M would increase the number of vertices of i and decrease the number of vertices. That contradicts the tie-breaking rule of the mechanism.

Now, let's assume that the mechanism breaks ties in favor of j over i. To arrive at a contradiction we will consider that subpath of $M \Delta M'$ that leaves V_i for the last time and ends at the end of the path. The subpath we are considering always starts with an edge of M'. The reason for that is the following: The first time we leave V_i, we leave with an edge of M' because either the first edge of $M \Delta M'$ is an external one, or we go through an even number of edges inside V_i and leave with an edge of M'. Then we enter a subpath of an agent belonging to the other part of the bipartition and we exit with an edge of M and enter an agent which could either i or some other agent that belongs to the first part of the bipartition, and we exit with an edge of M', and so on. It is the same kind of argument that we used in the previous case. We know therefore that the subpath we are considering always starts with an edge of M'. As we see in the figure, agent i controls the black nodes, and the edge in question is (v_3, v_4)

and it is an edge of M', as we have shown.

Figure 4.7: $|M_{ii}| = |M'_{ii}|$ and $|M| = |M'|$



Now, let's consider another matching, namely M''. M'' is M in the subpath we are considering and M' everywhere else. M'' has the same overall cardinality with M' and maximizes all internal edges. The only difference with M' is that it matches one more vertex of j and one less vertex of i. That contradicts the way our mechanism broke ties when agent i hid some of his vertices.

□

Also, we can reduce $MATCH_\Pi$ to the maximum weighted matching problem and prove that it can be executed in polynomial time.

Theorem 4.3. *$MATCH_\Pi$ can be executed in polynomial time.*

We have seen good qualities so far: Strategy-proofness and polynomial complexity. We are left to prove its approximation ratio. As we have seen in a previous chapter, the lower bound for deterministic algorithms is 2, even for 2 players. If we restrict our attention to two players we see that $MATCH_\Pi$ hits the lower bound. The reason is simple; our mechanism is maximal for two players. Since the players are in different parts of the bipartition no external edge is excluded. So, if a maximum matching contained an edge (x, y), then our mechanism would at least match one of x,y because if it didn't, it would be possible to include the edge in the matching and increase the cardinality which contradicts the definition of the mechanism.

Corollary 4.4. *Let $N = \{1, 2\}$. Then, $MATCH_{(\{1\},\{2\})}$ is strategyproof and provides a 2-approximation with respect to social welfare.*

The bad news is that our approximation ratio is way worse if we let many players in the mechanism. That is expressed in the theorem below.

Theorem 4.5. *$MATCH_{\Pi}$ does not provide a constant approximation ratio*

Proof. Let's consider the case of 3 players. Let $\Pi_1 = \{1, 2\}$ and $\Pi_2 = \{3\}$. Let's consider a graph which has a single edge between agent 1 and agent 2. Since agent 1 and 2 belong to the same side of the bipartition then our mechanism will not use that edge. Therefore the cardinality of the matching of $MATCH_{\Pi}$ is 0, whereas the cardinality of the maximum matching is 1. $\square$

In general, if we have many external edges between agents on the same side of the bipartition and very few between agents on different sides of the bipartition, $MATCH_{\Pi}$ would perform very poorly. However, even if our mechanism did hit the lower bound for deterministic mechanisms, that would still be bad since it would be at the same level of treating the problem as a simple optimization problem without taking into account incentives at all. Nevertheless, this mechanism is useful because it will help us build randomized mechanisms with good approximation ratios in the next chapter.

Chapter 5

A randomized mechanism for 2 agents: WEIGHT-AND-MATCH

We now turn to randomized mechanisms for two agents. That means that in at least one step, the mechanism makes a random choice. If we demand truthfulness in expectation and inclusion-maximality, the lower bound for the approximation ratio is 4/3. In this chapter we will present the WEIGHT-AND-MATCH mechanism which is analyzed in [22].

A general strategy for constructing a randomized mechanism that is both strategy-proof and as close to the optimal solution as possible, is the following: Select two deterministic mechanisms so that the first one has good incentive properties and bad approximation ratio and the second one has bad incentive properties and good approximation ratio. Flip a fair coin and if it comes heads, run the first mechanism and if it comes tails, run the other. One hopes that by randomizing between the two mechanisms, one gets the best qualities of both.

It is obvious that you will get an approximation ratio that is the weighted average of the two mechanisms depending on the coin. If we use a fair coin, both mechanisms have the same weight. So, you obviously improve in that regard in relation to the mechanism with a bad approximation ratio. The difficult part is the incentive property of the randomized mechanism. You must not "weaken" the strategyproof mechanism too much causing the overall randomized mechanism to lose the property of strategyproofness.

A deterministic mechanism that is strategy-proof but has a bad approximation ratio is the one we examined in the last chapter, $MATCH_{\Pi}$. We will first attempt to use $MATCH_{\Pi}$ in order to formulate a mechanism using the general strategy described above with the best approximation ratio possible.

5.1 FLIP-AND-MATCH

Algorithm 4 FLIP-AND-MATCH

Given a graph G, flip a fair coin.
If the outcome is heads, return $MATCH_{(\{1\},\{2\})}$
If the outcome is tails, choose a maximum cardinality matching, breaking ties in favor of a matching that maximizes the total number of internal edges and then arbitrarily.

Theorem 5.1. *Let $N = \{1, 2\}$. Then, FLIP-AND-MATCH provides a 4/3-approximation with respect to the social welfare.*

Proof. With probability 1/2 the mechanism chooses $MATCH_{(\{1\},\{2\})}$ which will give at least 1/2 of the optimal solution and with the same probability the mechanism chooses the optimal solution.

$$a \geq \frac{1}{2} \cdot \frac{1}{2} + \frac{1}{2} \cdot 1 \ = \frac{3}{4}$$

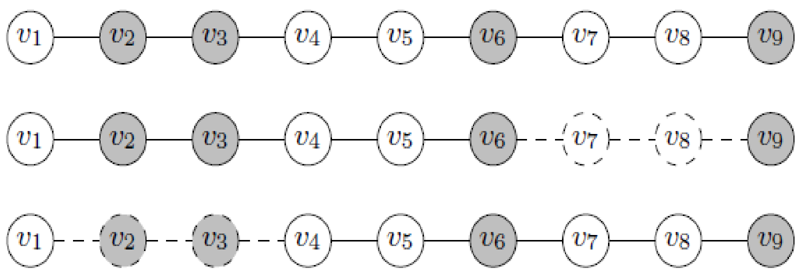
□

If this mechanism was truthful in expectation, we would have the best possible mechanism in our hands. However, that is not true.

Theorem 5.2. *Let $N = \{1, 2\}$. Then, FLIP-AND-MATCH is not truthful.*

Proof. All we need to do is to find a counter-example. Consider the graph in the figure. Agent 1 controls the white vertices and agent 2 controls with grey vertices.

Figure 5.1: Theorem 5.2 counter-example



The upper graph is the graph where both agents reveal all their vertices. $M_1 = MATCH_{(\{1\},\{2\})}$ would match $M_1 = \{(v_2, v_3), (v_4, v_5), (v_7, v_8)\}$ since these are all internal edges and we cannot include external edges without excluding some internal ones. Let M_2 be the maximum cardinality matching. M_2 would certainly leave out a vertex, either a white or a grey one, because the number of vertices in the graph is odd. We will therefore consider two cases:

1. M_2 leaves out a white vertex.

   That means that agent 1 gains 4 nodes from both M_1 and M_2, namely v_4, v_5, v_7, v_8. So, obviously the expected gain of agent 1 when revealing all of his vertices is 4. Now, let us examine the middle graph, where agent 1 hides (dashed lines) v_7 and v_8 and matches them internally. $M_1 = \{(v_2, v_3), (v_4, v_5)\}$ and M_2 would match all the vertices. Therefore, the gain of agent 1 will be 2 from M_1 and 3 from M_2 and thus agent's 1 expected gain will be 2.5. If you add the hidden vertices he matched on his own his total expected gain would 4.5, which is more than agent's 1 gain had he tell the truth.

2. M_2 leaves out a grey vertex.

   That means that agent 2 gains 3 nodes from M_2 and 2 nodes from M_1, and thus his expected gain in the upper graph is 2.5. Let us suppose that agent 2 hides (v_2, v_3) (lower graph). M_2 would match all the vertices and would give him a gain of 2 and M_1 would match non of his vertices and give him no gain. So, his overall expected gain when lying would be 3, which is more than the one in the upper graph.

$\square$

It is interesting to see that whatever constant probabilities we use to randomize between those two deterministic mechanisms, the randomized mechanism is still not truthful. For example, let p be an arbitrary probability $0 < p < 1$ and let us say we select M_1 with probability p and M_2 with probability $(1 - p)$. In the first case we examined, agent's 1 gain when telling the truth would still be 4, and his gain when lying would be $5 - p > 4$, since $0 < p < 1$. Finally, in the second case, agent's 2 gain when telling the truth would be $3 - a$ and his gain when lying would be $4 - a > 3 - a$.

So, we have seen that FLIP-AND-MATCH is not truthful and thus using the maximum cardinality matching as the bad-incentive mechanism, we cause the $MATCH_{\Pi}$'s contribution to incentives to weaken significantly. The solution is to use a mechanism with better incentive properties and worse approximation than the maximum cardinality matching. It doesn't mean that this new deterministic mechanism must be strategy-proof; it may very well be the case that is not strategy-proof but closer to strategy-proofness than the maximum cardinality matching mechanism.

5.2 WEIGHT-AND-MATCH

Algorithm 5 WEIGHT-AND-MATCH

1: Given a graph G, assign weights w to get G':

1. w=1 for internal edges
2. w=0.5 for external edges

2: Flip a fair coin

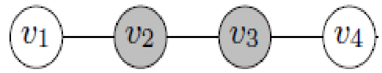
3: If the outcome is heads, return the maximum-weight matching with minimum cardinality

4: If the outcome is tails, return the maximum-weight matching with maximum cardinality

The WEIGHT-AND-MATCH mechanism is constructed with the same strategy we used before. The deterministic strategy-proof mechanism is the mechanism that returns the maximum-weight with minimum cardinality. Actually that mechanism is $MATCH_{\Pi}$. That is because of the fact that if you want to get maximum weight with as few edges as possible, you must include as many internal edges as possible. Obviously, the mechanism that does that is $MATCH_{\Pi}$. Now, the mechanism with maximum-weight and maximum cardinality is closer to optimal than $MATCH_{\Pi}$, but it is not optimal. We can demonstrate that with two examples.

In the first example,we see that $MATCH_{\Pi}$ will select (v_2, v_3) whereas the maximum-cardinality matching will select (v_1, v_2) and (v_3, v_4). In this case, the maximum-

Figure 5.2: Example 5.2.1



weight/maximum-cardinality is the same with the maximum matching. We know that $w(v_2, v_3) = 1$ since (v_2, v_3) is an internal edge and $w(v_1, v_2) + w(v_3, v_4) = 1$ since (v_1, v_2) and (v_3, v_4) are external edges. Therefore both matchings have the same weight, and so the maximum-weight/maximum cardinality matching will choose the matching with cardinality 2, namely $\{(v_1, v_2), (v_3, v_4)\}$.

Figure 5.3: Example 5.2.2

However, in the second example, the maximum-weight/maximum-cardinality matching differs from the maximum cardinality matching and seems to give more priority to internal edges. That could be said intuitively to have better incentive properties than the maximum matching although it is not enough to be strategy-proof. The maximum matching in this case is $\{(v_1, v_2), (v_3, v_4), (v_5, v_6)\}$ which has a total weight of 1.5. The maximum-weight/maximum-cardinality matching will obviously be $\{(v_2, v_3), (v_4, v_5)\}$ which has a weight of 2. Therefore, this mechanism sacrifices optimality for internal edges in this case, which causes his approximation to be worse than the optimal but better than $MATCH_\Pi$ overall.

Theorem 5.3. *Let $N = \{1, 2\}$. Then, WEIGHT-AND-MATCH provides a 3/2-approximation with respect to the social welfare.*

Proof. Let M be the maximum-cardinality matching, M_1 be the maximum-weight/minimum-cardinality matching and M_2 be the maximum-weight/maximum-cardinality matching. We use the symmetric difference $M \Delta M_1$ to aid our proof. As we have seen in chapter 11 and lemma 11.1, the symmetric difference of two matchings consists of vertex-disjoint paths or cycles.

We define m_1 to be the number of edges of M that either belong also to M_1, or belong to paths or cycles of even length in $M \Delta M_1$, m_3 to be the number of edges of M that belong to paths of length 3 in $M \Delta M_1$ and m_5 to be the number of edges of M that belong to paths of odd length that is greater or equal to 5 in $M \Delta M_1$. With these definitions, we basically covered all of M:

$$|M| = m_1 + m_3 + m_5$$

We observe that m_1 is the number of edges of M_1 that either also belong to M, or that belong to paths or cycles of even length in $M \Delta M_1$. The reason is of course that what is common for both matchings has obviously the same cardinality, and the paths and cycles of even length in $M \Delta M_1$ have all their vertices covered by both M and M_1. Since M is maximum-cardinality and M_1 is maximum-weight/minimum-cardinality, M has strictly greater cardinality than M_1, and all the paths of odd length in $M \Delta M_1$ begin and end with an edge of M. So, if m_3 are the number of edges of M in paths of length 3 in $M \Delta M_1$, then $m_3/2$ is the number of edges of M_1 in those paths because M gets the first and last edge of the path and M_1 gets the second edge of the path, in every such path (the same as the first example before). If we consider paths of length 5 for example, then M_1 gets the second and forth edge in the path. So, he gets 2/3 the number of edges M gets in that path (second example). As we move on to odd paths of greater length the proportion of M_1 edges in relation to M's increases.

Therefore,

$$|M_1| \geq m_1 + \frac{m_3}{2} + \frac{2m_5}{3}$$

We know that M_1 and M_2 are maximum weight matchings and M_2 has greater cardinality than M_1. Therefore, we can say that $|M_2| \geq |M_1|$. Nevertheless, we can make an improvement on that lower bound. Let's consider paths of length 3 in $M \Delta M_1$. As we said in the example above, in this case M_2 is identical with M, because M_1 and M have the same weight but obviously M has the greater cardinality. Of course, one could claim that the middle edge need not be an internal one. If it is not, M_1 would not have chosen it and the path would not be in $M \Delta M_1$. So, we have

$$|M_2| \geq m_1 + m_3 + \frac{2m_5}{3}$$

if we combine those inequalities, we find that the expected cardinality of WEIGHT-AND-MATCH

$$\frac{1}{2}(|M_1| + |M_2|) \geq m_1 + \frac{3m_3}{4} + \frac{2m_5}{3} \geq \frac{2}{3}(m_1 + m_3 + m_5) = \frac{2}{3}|M|$$

□

If we examine the graph in the second example before, we can see that the analysis of the theorem is tight. The maximum matching will return all the nodes, namely 6. Both max-weight/min-cardinality and max-weight/max-cardinality matchings will return $\{(v_2, v_3), (v_4, v_5)\}$ so the expected cardinality of WEIGHT-AND-MATCH is 4. We will now prove that WEIGHT-AND-MATCH is truthful.

Theorem 5.4. *Let $N = \{1, 2\}$. Then, WEIGHT-AND-MATCH is truthful in expectation.*

Proof. Let G be the input graph. Then M_1 is the max-weight/min-cardinality matching and M_2 is the max-weight/max-cardinality on G. If agent 1 hides some of its vertices, G' is induced from G. We define M_3 and M_4 the max-weight/min-cardinality and max-weight/max-cardinality matchings respectively on the graph G' augmented by the internal edges of agent 1. Lastly, we define u as the utility of agent 1 and $wgt(M)$ as the total weight of matching M. We will only examine the incentives of agent 1 without loss of generality.

Claim 1

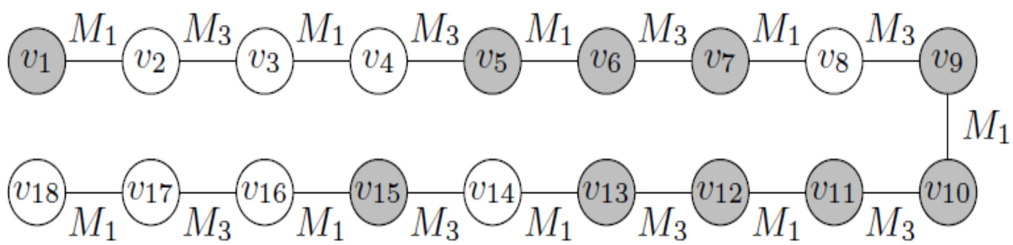
$$u(M_3) = u(M_1) - 2(wgt(M_1) - wgt(M_3))$$

We will consider again the symmetric difference $M_1 \Delta M_3$ which is composed, as we already discussed, of vertex-disjoint paths or cycles. We will assume only one such path without losing generality because if we prove this equality for one arbitrary path, the equality will hold for all the vertex-disjoint paths and therefore will hold in total since outside $M_1 \Delta M_3$ there is no difference in utility between the two matchings.

Slightly changing notation from previous chapters we denote $n_{11}(M)$, $n_{22}(M)$ and $n_{12}(M)$ the number of agent's 1 internal edges, agent's 2 internal edges and the number of external edges, respectively. We will first show that $n_{22}(M_1) = n_{22}(M_3)$. If we remember that the maximum-weight/minimum-cardinality matching is actually $MATCH_{\Pi}$ from the previous chapter, then we have already shown this in a previous proof. However, since no formal proof was presented for this, we will show it again from the perspective of a weighted graph.

Odd cycles cannot exist in a symmetric difference between two matchings and even cycles have both matchings covering all the vertices. Therefore, we can exclude cycles altogether. We will examine the subpaths of agent 2 in our path in $M_1 \Delta M_3$. We claim that all the subpaths of agent 2 have an even number of edges and if that is the case, then $n_{22}(M_1) = n_{22}(M_3)$ immediately follows.

Figure 5.4: Theorem example



Let's assume that the extremes vertices of this subpath have a degree of 2 and that the subpath has an odd number of edges. Let's say that M_1 has one more internal edge than M_3. That means that M_3 has 1 less unit of weight than M_1. However, since the extreme vertices have a degree of 2, that means that M_3 gains back the 1 unit of weight through 2 external edges. But then M_3 will have the same weight as M_1 but larger cardinality. That cannot be the case since M_3 is a max-weight/min-cardinality matching and so the subpath cannot be of odd length. The same holds if M_3 has one more internal edge than M_1. If we assume that at least one of the extreme vertices has a degree of 1, that means that the matching with the fewest internal edges will also have less weight since there is no external edge to make up for its internal weight loss. But then, that matching cannot be maximum-weight mathing since it can switch to the other and increase its weight. Therefore, as before, the subpath of agent 2 cannot have an odd number of edges. So, $n_{22}(M_1) = n_{22}(M_3)$. We can see in the Figure an example of this case where the white vertices belong to agent 1 and the grey vertices to agent 2.

In general it holds that $u(M) = 2n_{11}(M) + n_{12}(M)$ and $wgt(M) = n_{11}(M) + n_{22}(M) + n_{12}(M)/2$. Therefore, since $n_{22}(M_1) = n_{22}(M_3)$, it holds that

$$\begin{aligned} u(M_3) &= 2n_{11}(M_3) + n_{12}(M_3) \\ &= 2n_{11}(M_3) + n_{12}(M_3) + 2n_{22}(M_3) - 2n_{22}(M_1) \\ &= u(M_1) + 2(wgt(M_1) - wgt(M_3)) \end{aligned}$$

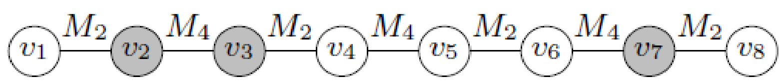
Claim 2

$$u(M_4) \leq u(M_2) + 2(wgt(M_2) - wgt(M_4))$$

In this case we will also consider the symmetric difference, namely $M_2 \Delta M_4$ and we can assume that it is composed of only one path. Since M_2 is a maximum weight matching on G, it holds that $wgt(M_2) \geq wgt(M_4)$. We observe that since the internal vertices are mathced by both matchings in a path, the extreme vertices play a crucial role for the utitlity of agent 1. Because of that, we will use a special notation. For example, a path of the type $a22b$ is a path who has the first vertex belonging to agent 1 with an edge of M_2 and the last vertex belonging to agent 2 with also an edge of M_2. So, the first two characters are about the starting point of the path and the last two are about the ending point of the path; a and b are are about agent 1 and 2 respectively; 2 and 4 are about M_2 and M_4 respectively. We will consider 3 different cases.

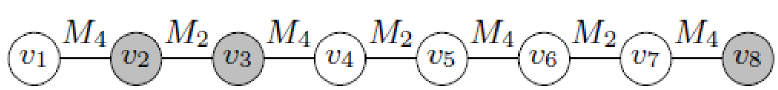
Firstly, if $M_2 \Delta M_4$ is a cycle or a path of the types $a22a$, $a24a$, $a42a$, $a22b$, $a24b$, $b22b$, $b24b$, $b42b$, $b44b$. In all these cases it holds that $u(M_4) \leq u(M_2)$. The reason is that if there are extreme vertices of agent 1 and at least one of agent's 1 extreme vertices is matched by M_2, then there is at most 1 other extreme vertex that is matched by M_4. Since $wgt(M_2) \geq wgt(M_4)$, the inequality holds. In the Figure below there is an example of this case. If we don't take into account the extreme vertices, both M_2 and M_4 match the 3 white vertices of agent 1, but since M_2 matches both extreme vertices of agent 1, M_2 adds to agent's 1 utility 2 more than M_4.

Figure 5.5: First type of path, $a22a$



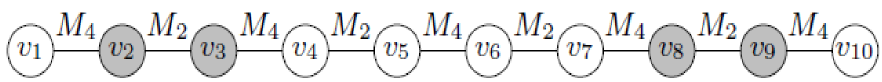
Secondly, let's assume that our path is either $a42b$ or $a44b$. The first and last vertex belongs to different agents and therefore we have an odd number of external edges. All internal edges contribute 1 and all external edges contribute 1/2 to $wgt(M_2) + wgt(M_4)$. Therefore, this sum is not an integer since we have an odd numver of external edges. Also, one of the $wgt(M_2)$, $wgt(M_4)$ is a non-integer and the other one is. So, since $wgt(M_2) \geq wgt(M_4)$, we conclude that $wgt(M_2) - wgt(M_4) \geq 1/2$. This, combined with the fact that $u(M_2) + 1 = u(M_4)$, shows us that the desired inequality holds. We have an example below in the Figure, that demonstrates exactly this argument.

Figure 5.6: Second type of path, $a44b$



Lastly, let's assume that our path is of the type $a44a$. That means that both extreme vertices of agent 1 belong to M_4. So, $u(M_4) = u(M_2) + 2$. This time since the extreme vertices belong to the same agent, we have an even number of external edges and so the $wgt(M_2) + wgt(M_4)$ is an integer. Therefore, the difference of weights is an integer like 0 or more. Can the matchings have the same weight? No, because M_4 has one more edge than M_2 and since M_2 can switch to M_4 to keep its weight but increase its cardinality, we can infer that M_2 is not a max-weight/max-cardinality matching, which is a contradiction. Therefore $wgt(M_2) \geq wgt(M_4) + 1$, which leads us to prove the inequality. In the figure below there is an example of this case as well.

Figure 5.7: Third type of path, $a44a$



Let's use our claims now to prove strategyproofness. If we add up $u(M_4) \leq u(M_2) + 2(wgt(M_2) - wgt(M_4))$ and $u(M_3) = u(M_1) - 2(wgt(M_1) - wgt(M_3))$ and divide by 2, we have that the expected utility of agent 1 is

$$\frac{1}{2}(u(M_1) + u(M_2)) \geq \frac{1}{2}(u(M_3) + u(M_4))$$

$\square$

An interesting thing to see is that this mechanism cannot gain a better approximation ratio without losing its strategy-proofness with just a change in the probabilities with which it randomizes over the two deterministic mechanisms. Let's look at the example in the last figure and let's say that that is the real graph G while ignoring all the M_4's and M_2's in the figure. Both max-weight/min-cardinality and max-weight/max-cardinality will choose $\{(v_2, v_3), (v_4, v_5), (v_6, v_7), (v_8, v_9)\}$. So whatever the probabilities of choosing the deterministic mechanisms agent's 1 gain will be 4.

Let's say that we choose the max-weight/min-cardinality matching with $(0.5 - \epsilon/4)$ probability and the max-weight/max-matching with $(0.5 + \epsilon/4)$ probability, where $\epsilon > 0$. Let's say that agent 1 hides v_5, v_6 and matches them internally. The subgraph that is induced has two disconnected components and in each component the mechanism will behave identically. The max-weight/min-cardinality matching will select no vertex of agent 1, but the max-weight/max-cardinality matching will select both vertices from each component, totaling in 4 agent's 1 vertices. So, the expected gain from agent 1 when lying is $u(M') = 2 + (0.5 + \epsilon/4)4 = 4 + \epsilon > 4 = u(M)$. Therefore, even if we slightly change the probabilities in order to improve the approximation ratio, we will lose the property of strategyproofness.

Lastly, just as expected, our mechanism is executed in polynomial time. So, we have a computationally efficient mechanism, which is strategy-proof in expectation and has an approximation ratio of 3/2. The basic limitation of this mechanism is that it is useful in settings where only two players are involved. So, in the next chapter we will see a mechanism that works for many players.

Chapter 6

A randomized mechanism for many agents: MIX-AND-MATCH

This mechanism is a natural extension of the $MATCH_\Pi$ mechanism. We will confront the greatest weakness and at the same time greatest strength of this deterministic mechanism with randomization and we will solve the problem of non-constant approximation ratio [13]. First, as a reminder, let's see the example used before to see why the $MATCH_\Pi$ mechanism has a non-constant approximation ratio.

Let's consider the case of 3 players. Let $\Pi_1 = \{1, 2\}$ and $\Pi_2 = \{3\}$. Let's consider a graph which has a single edge between agent 1 and agent 2. Since agent 1 and 2 belong to the same side of the bipartition then our mechanism will not use that edge. Therefore the cardinality of the matching of $MATCH_\Pi$ is 0, whereas the cardinality of the maximum matching is 1.

So, the problem lies in the fixed bipartition and its external edges limitation. The mechanism that follows deals with that directly using randomization.

Algorithm 6 MIX-AND-MATCH

1: **Mix**: Construct a random bipartition $\Pi = (\Pi_1, \Pi_2)$ of the agents by independently flipping a fair coin for each agent to determine whether the agent is in Π_1 or in Π_2.
2: **Match**: Apply $MATCH_\Pi$ to the given graph, where Π is the bipartition constructed in the previous step

Obviously, this mechanism is universally strategy-proof. Regardless of the choice of Π, all agents have no incentive to lie, since $MATCH_\Pi$ is strategyproof. More formally, we define a randomized mechanism as a probability distribution over deterministic mechanisms and that a ranomized mechanism is universally strategyproof if all these deterministic mechanisms are strategyproof. Therefore, since $MATCH_\Pi$ is strategyproof, the conclusion is obvious.

Let's examine what happens to the approximation ratio using the example in which $MATCH_\Pi$ performed poorly. We can work out that agent 1 and agent 2 who have the external edge between them will belong to the same side of the bipartition with probability 1/2 and in different sides of the bipartition with probability 1/2 as well. That means that with probability 1/2 the mechanism returns no edges (as does $MATCH_\Pi$) and with probability 1/2 will return one edge since the external edge on those bipartitions will be between different sides of Π. The optimal matching has cardinality of 1 and the expected cardinality given by the MIX-AND-MATCH matching is 1/2. It seems that by using randomization we turned the worse-case non-constant approximation to an approximation ratio of 2.

However, that was just an example. $MATCH_{\{1\},\{2\}}$ sacrifices 2 external edges for 1 internal edge and therefore it has an approximation ratio of 2. In MIX-AND-MATCH we also have the problem of removing external edges because the players are more than 2. So, in general one might think that the approximation ratio of MIX-AND-

MATCH will be even worse than $MATCH_{\{1\},\{2\}}$. But, that is not the case. The idea is that half of the time those external edges that would be sacrificed are already removed by the choice of the bipartition.

Theorem 6.1. *For any number of agents, MIX-AND-MATCH is (universally) strategyproof and provides a 2-approximation with respect to social welfare.*

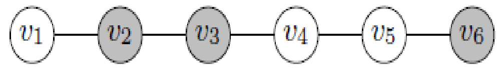
Proof. At first, using the maximum matching M^* we will contruct M' which is a matching that we can easily show that it has an approximation of 2 on average. We finalize the proof by showing that the cardinality of $MATCH_\Pi$ is greater than the cardinality of M' when restricted by Π. That means that the cardinality of the matching given by MIX-AND-MATCH on average is greater than that of M'.

Let ${M_i}^{**}$ be a maximum cardinality matching on V_i and $M^{**} = \cup_{i \in N} {M_i}^{**}$. We consider the symmetric difference $M^* \Delta M^{**}$. For each vertex-disjoint path of $M^* \Delta M^{**}$ the number of internal edges of M^{**} is either equal or greater than M^*. If it is equal, we add M^*'s edges to M', else, we add M^{**}'s edges to M'. From construction M' maximizes the total number of internal edges.

Since M^* is maximum matching, then every path in $M^* \Delta M^{**}$, M^* has at most one more edge than, and at least the same number of edges as, M^{**}. All external edges on the path are from M^* , so if the edges from M^{**} are taken for M' then the number of internal edges gained relative to M^* is at least the number of external edges lost minus one. In the worst case M' has two fewer external edges for each extra internal edge relative to M^*. To understand this better, let's use two examples.

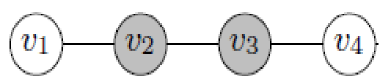
$M^* = \{(v_1, v_2), (v_3, v_4), (v_5, v_6)\}$ and $M^{**} = \{(v_2, v_3), (v_4, v_5)\}$. Since M^{**} has 2 internal edges and M^* has not, M' gets M^{**}'s edges. So, M' gains 2 extra internal

Figure 6.1: $M^*\Delta M^{**}$ example



edges and loses 3 external edges. In the figure below, we see another example where $M' = \{(v_2, v_3)\}$ and $M^* = \{(v_1, v_2), (v_3, v_4)\}$. M' has exactly 2 fewer external edges for 1 internal edge that it gains. We can easily see how this pattern can be generalized, if we observe that in each subpath of V_j in $M^*\Delta M^{**}$, M' will have at most one more internal edge than M^*, and at most two less external edges than M^*. We need in general many such subpaths to complete one vertex-disjoint path of $M^*\Delta M^{**}$. We see that in the figure below, we essentially have a subpath of the figure above and the property holds for both.

Figure 6.2: $M^*\Delta M^{**}$ example



So, we have

$$\sum_{i\in N} (|M'_{ii}| - |M^*_{ii}|) \geq \frac{1}{2}\sum_{i\in N}\sum_{j>i} (|M^*_{ij}| - |M'_{ij}|)$$

$$\Rightarrow \sum_{i\in N} |M'_{ii}| + \frac{1}{2}\sum_{i\in N}\sum_{j>i} |M'_{ij}| \geq \sum_{i\in N} |M^*_{ii}| + \frac{1}{2}\sum_{i\in N}\sum_{j>i} |M^*_{ij}|$$

Since $MATCH_\Pi$ is the maximum matching given the restriction of external edges, then it holds that

$$|M^{\Pi}| = \sum_{i \in N} |M^{\Pi}_{ii}| + \sum_{i \in \Pi_1} \sum_{j \in \Pi_2} |M^{\Pi}_{ij}| \geq \sum_{i \in N} M'_{ii} + \sum_{i \in \Pi_1} \sum_{j \in \Pi_2} |M'_{ij}|$$

So, we observe at last that in exactly half of the bipartitions, two specific agents belong to different sides of Π. That means that the first inequality shows that the average cardinality of MIX-AND-MATCH is greater or equal to the average cardinality of M'. From there, the result follows.

$$\begin{aligned}
\sum_{\Pi} \left(1/2^n \cdot |M^{\Pi}|\right) &\geq \sum_{i \in N} |M'_{ii}| + 1/2 \sum_{i \in N} \sum_{j > i} |M'_{ij}| \\
&\geq \sum_{i \in N} |M^*_{ii}| + 1/2 \sum_{i \in N} \sum_{j > i} |M^*_{ij}| \geq 1/2 \cdot |M^*|
\end{aligned}$$

□

Chapter 7

A randomized mechanism for many agents: BONUS

In this chapter we will change perspective and try to use distribution information about patients to increase the efficiency of the mechanism. We will introduce some elements of random graph theory and then we will see that we can have an efficient allocation of kidneys in our specific random graphs. Moreover, we have an 'almost' efficient allocation which is individually rational and we will derive the BONUS mechanism [20] that makes it a Bayes-Nash equilibrium for the hospitals to reveal all their incompatible pairs with cardinality very close to optimal. The most important difference that this chapter has from the previous chapters is that it allows not only 2-way exchanges but also 3-way exchanges as well.

7.1 Random Graphs

We will now look at some basic Random Graph Theory. A Random graph $G(m, p)$ is a graph with m nodes and between two different nodes there exists an undirected edge with probability p (p is a non-increasing function of m). A bipartite random graph $G(m,m,p)$ has two disjoint sets of size m each and there exists an edge with probability p between the two sets. For any graph theoretic property Q there is a probability that a random graph G satisfies Q, denoted by $Pr(G \models Q)$. A matching in an undirected graph is a set of edges for which no two edges have a node in common. A matching is nearly perfect if it matches (contains) all but at most one nodes in the graph, and perfect if it matches all nodes. A very useful theorem follows [65] that essentially states that if a random graph is large enough there exists a perfect matching for it.

Theorem 7.1 (Erdos-Renyi). *Let Q be the property that there exists a nearly perfect matching. For any constant p*

1. $Pr(G(m,p) \models Q) = 1 - o(1)$.

2. $Pr(G(m,m,p) \models Q) = 1 - o(1)$.

We will say for any property Q where $Pr(G \models Q) = 1 - o(1)$. that Q holds in almost every large graph G. It's important to note that this theorem holds in an even stronger version. Let $r(m) = \frac{lnm}{m}$ be a threshhold function. If $p = p(m)$ is such that $r(m) = o(p(m))$ then the probability a nearly perfect matching exists converges to 1, and if $p(m) = o(r(m))$, the probability a nearly perfect matching exists converges to 0.

We will now look at the specific random graphs we will actually use later. Recall that there are two compatibility tests between a donor and a patient: the blood compatibility test and the tissue-type compatibility test. In the first test, we check if the blood type of the donor does not have proteins that the patient's blood type does not have-i.e ABO compatibility. That means that an O type can give to all, an

A type to A and AB, a B type to B and AB and an AB type only to itself. The probability that a random person's blood is X is given by $\mu_X > 0$. In the second test, we must not have what is called a positive crossmatch. Each patient has a level of Percentage Reactive Antibodies (PRA) which determines the likelihood that the patient will be incompatible with a random donor. For simplicity, we assume only two levels of PRA: high (H) and low (L). The probability that a patient p with PRA L (H) and a donor are tissue type incompatible is given by γ_L (γ_H). To create a random compatibility graph, we first draw the incompatible pairs,i.e donors and their intented patients, as the vertices and then the edges as compatibilities between pairs according to the distribution assumptions.

Definition 7.1. *A random (directed) compatibility graph of size m, denoted D(m), consists of m incompatible patient-donor pairs, and a random edge is generated between every donor and each patient compatible with that donor. Hence, such a graph is generated in two phases:*

1. *Each node/incompatible pair in the graph is randomized as follows. A patient p and a potential donor d are created with blood types drawn independently according to the probability distribution* $\mu = (\mu_X)_{X \in \{A,B,AB,O\}}$ *. The PRA of patient p, denoted by* $\gamma(p)$*, is also randomized (L with probability v and H with probability 1 - v). A number z is drawn uniformly from* $[0,1]$ *and* (p,d) *forms a new node if and only if p and d are blood type incompatible or p and d are blood type compatible but* $z \leq \gamma(p)$ *(so p and d are tissue type incompatible).*

2. *For any two pairs* $v_1 = (p_1, d_1)$ *and* $v_2 = (p_2, d_2)$*, there is an edge from* v_1 *to* v_2 *if and only if* d_1 *and* p_2 *are ABO compatible and also tissue type compatible (*d_1 *is tissue type compatible with* p_2 *with probability* 1bﬄ″$\gamma(p_2)$*).*

7.2 Efficient Allocations in Large Random Graphs

We will now see if there exists an efficient allocation in our Kidney exchange graph without taking into account the incentives of hospitals at all. The first thing we need to know is what sizes are the groups of incompatible pairs with a specific blood type pairs. For example A-O pairs, that is pairs who have a patient of blood type A and a donor of blood type O, may have a different size than O-A pairs. The reason is that we will match certain groups through 2-way or 3-way exchanges and we need to see if there exist incompatible pairs that were left out after those exchanges, so that we can match them with other blood-type groups.

Lemma 7.1. *In almost every large $D(m)$:*

1. *For all $X \in \{A, B, AB\}$ the number of O-X pairs is larger than the number of X-O pairs.*

2. *For all $X \in \{A, B\}$ the number of X-AB pairs is larger than the number of AB-X pairs.*

3. *The absolute difference between the number of A-B pairs and B-A pairs is $o(m)$. Consequently this difference is smaller than the number of pairs of any other pair type.*

This lemma is proved technically using Chernoff bounds but the idea behind the proof is simple. The probability for us to find an O-B pair and a B-O pair in general, is identical. The difference is that for the B-O pair to make the cut for our graph, the patient and donor must also satisfy the extra condition of being tissue-type incompatible. So, we have less probability to find a B-O pair than an O-B pair in our graph and so in almost every large graph the cardinality of O-B pairs is larger than the one of B-O pairs. With this reasoning the first two parts of the lemma can be proved.

With this lemma in mind, we can partition the set of patients $\mathcal{P}$ to overdemanded $\mathcal{P}^{\mathcal{O}}$, underdemanded $\mathcal{P}^{\mathcal{U}}$, self-demanded $\mathcal{P}^{\mathcal{S}}$ and reciprocally demanded $\mathcal{P}^{\mathcal{R}}$ pairs. The overdemanded pairs are the ones who offer a kidney which is in larger demand than the kidney they seek. Those pairs are the pairs in which the patient and donor have different blood types and the donor is blood-type compatible to donate a kidney to its patient but not tissue-type compatible. Generally if someone is blood-type compatible to donate a kidney to someone else, then he's able to donate to more blood-types than the blood-type he is compatible with. The underdemanded pairs are the pairs which seek a kidney that is in greater demand than the kidney they offer. The self-demanded pairs are the X-X pairs and the reciprocally demanded pairs are the A-B and B-A pairs.

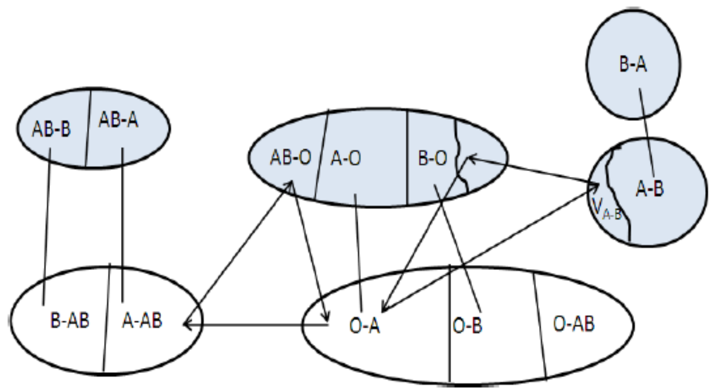
Using lemma 7.1, the Erdos-Renyi theorem and its extension to l-partite graphs, we can prove the following proposition which gives us an efficient allocation of kidneys in our random graph. That allocation is evident in the figure below in which the colored types are the overdemanded types which are all mathced, leaving only underdemanded types unmatched. Since an overdemanded pair can help at most 2 underdemanded pairs to be matched and only AB-O pairs can do that, we know that our allocation is efficient.

Proposition 7.2. *Almost every large $D(m)$ has an efficient allocation that requires exchanges of no more than size 3 with the following properties:*

1. *Every selfdemanded pair X-X is matched in a 2-way or a 3-way exchange with other selfdemanded pairs (no more than one 3-way exchange is needed, in the case of an odd number of X-X pairs).*

2. *Either every B-A pair is matched in a 2-way exchange with an A-B pair or every A-B pair is matched in a two way exchange with a B-A pair.*

Figure 7.1: An efficient allocation



3. *Let X, Y $\in$ A, B and X = Y. If there are more Y-X than X-Y then every Y-X pair that is not matched to an X-Y pair is matched in a 3-way exchange with an O-Y pair and an X-O pair.*

4. *Every AB-O pair is matched in a 3-way exchange with an O-A pair and an A-AB pair.*

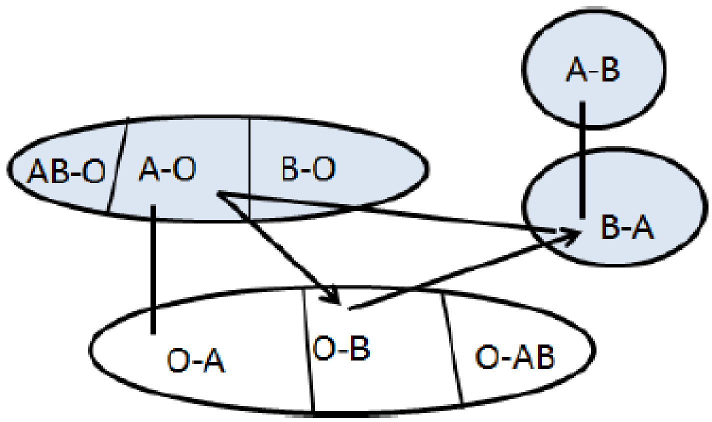
5. *Every overdemanded pair X-Y that is not matched as above is matched to an underdemanded Y-X pair.*

7.3 Individually Rational Allocations

In the previous section we saw how efficient allocations look like. Now, we have to find an allocation that is individually rational without changing the efficient allocation too much. For example if a hospital matches internally an A-O pair with A-A, B-A or an AB-A pair, then we would have 2 transplants instead of the 4 transplants we could have according to the efficient allocation before.

Let's consider the so called "unbalanced" 3-way exchange (A-O,O-B,B-A) along with some 2-way exchanges as seen in the figure below. This exchange is done internally in a hospital and it uses overdemanded pairs inefficiently. If we wanted to achieve individual rationality by attempting to include the internal allocations to the efficient allocation, then we would have a problem, since potentially we would need to match more O-B pairs than the total number of B-O pairs. Therefore, we would lose efficiency.

Figure 7.2: An efficient allocation



So, individually rational allocations may contain more underdemanded pairs of a specific kind than its reciprocally overdemanded type. However, this is not likely to happen, since hospitals are not "big" enough to have those kinds of "unbalanced" 3-way exchanges. We say that a hospital size c is regular if by randomly choosing an internal allocation that maximizes the number of underdemanded pairs, for any underdemanded type X-Y, the expected number of matched X-Y pairs is less than the expected number of overdemanded pairs in its pool. With that assumption for hospital sizes, we can get quite close to an efficient individually rational allocation as the next theorem suggests.

Theorem 7.3. *Suppose every hospital size is regular and bounded by some $c > 0$ and let $\epsilon > 0$. In almost every large graph $D(H_n)$ there exists an individually rational allocation using exchanges of size at most 3, which is at most $\mu_{AB-O}m + \mu_{A-B}m$ smaller than the efficient allocation, where m is the number of pairs in the graph.*

As suggested by the theorem, most of the efficiency loss comes from 2-way exchanges between AB-O pairs and O-AB pairs. However, it is found that the efficiency loss in practice in only about one percent. That shows that using random graphs with these particular assumptions, individually rational allocations are much closer to the efficient allocations than before.

7.4 Kidney exchange mechanisms in the Bayesian Setting- Bonus Mechanism

In this setting we will weaken our incentive criterion from strategy-proofness to $\epsilon-$Bayes-Nash equilibrium. We will not attempt to define it formally since it requires same basic Game Theory first. However, this means that if a hospital knows that the other hospitals will reveal all their pairs, that hospital has a good incentive to reveal all its pairs too.

We need to prevent hospitals from withholding overdemanded pairs from the mechanism. One way to to do that is determining beforehand which underdemanded pairs will be matched using a lottery. We will not give the details of the lottery; we will just present the mechanism along with the main result.

Theorem 7.4. *Let H_n be a set of hospitals. If every hospital size is strongly regular, the truth-telling strategy profile is an $\epsilon(n)$-Bayes-Nash equilibirum in the game induced* *by the Bonus mechanism, where* $\epsilon(n) = o(1)$*. Furthermore for any* $\epsilon > 0$*, the efficiency loss under the truth-telling strategy profile in almost every* $D(H_n)$ *is at most* $\mu_{AB-O}m+ \epsilon\mu_{Abffi''B}m$*, where* m *is the number of pairs in the pool.*

Algorithm 7 The Bonus Mechanism

1: [Input]: a set of hospitals $H_n = \{1, ..., n\}$ and a profile of incompatible pairs $(B_1, B_2, ..., B_n)$, each of a strongly regular size.

2: [Match selfdemanded pairs]: find a maximum allocation, M_S in the graph induced by all selfdemanded pairs B_{H_n} .

3: [Match A-B and B-A pairs]: for each hospital h choose randomly an allocation $M_h \in {\mathcal{M}_{\mathcal{P}^{\mathcal{R}}}}^{B_h}$. Find a maximum allocation M_R in the graph induced by A-B and B-A pairs among those that maximize the number of matched pairs in $\cup_{h\in H_n}\tau(M_h(B_h), \mathcal{P}^{\mathcal{R}})$.

4: [Match overdemanded and underdemanded pairs]: Partition the set of hospitals into two sets $H_n^1 = \{1, ..., \frac{n}{2}\}$ and $H_n^2 = \{\frac{n}{2} + 1, ..., n\}$. For each underdemanded type $X - Y \in \mathcal{P}^{\mathcal{U}}$ and for each $j = 1, 2$. Then, using the underdemanded lottery procedure, construct a subset $S_h(X - Y)$ one for each hospital in $h \in H_{n^j}$.

- Set $\theta_j(Y - X) = |\tau(B_{H_{n^{3bffi''j}}}, Y - X)|$ to be the number of Y-X pairs in the set $B_{H_{n^{3bffi''j}}}$. Then, using the underdemanded lottery procedure with the inputs $(B_h)_{h\in H_{n^j}}$, $\theta_j(Y - X)$ and X-Y, construct a subset $S_h(X - Y)$ one for each hospital in $h \in H_{n^j}$.
- Find a maximum allocation M^j_{X-Y} in the subgraph induced by the sets of pairs $\cup_{h\in H_{n^j}} S_h(X - Y)$ and $\tau(B_{H_{n^{3-j}}}, Y - X)$.

5: [Output]: Let $M_U = \cup_{j=1,2} \cup_{X-Y\in\mathcal{P}^{\mathcal{U}}} M^j_{X-Y}$. Output $M_S \cup M_R \cup M_U$.

As we see, this mechanism will have one percent efficiency loss according to the distributions used. However, the disadvantage of this mechanism and its analysis is that it makes crucial distribution assumptions that might not hold in a realistic setting. For example, graphs may not be as dense as was assumed. Actually, it was found later that most patients participating in kidney exchange are sensitized patients [19], meaning they have large probabilities to reject a random kidney. However, this model will be more accurate when the Kidney Exchange Program will expand on a national level. Using similar distribution assumptions, one can obtain an almost efficient allocation demanding individual rationality using only 2-way exchanges [21].

Chapter 8

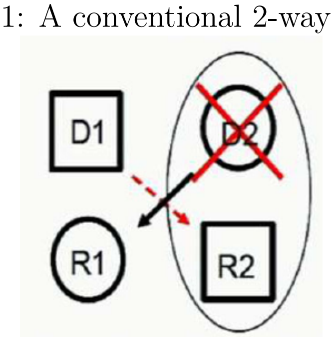
Conclusions - Future Work

At least until recently, the algorithm used for conducting kidney exchange had no consideration of incentives whatsoever. It has been observed that because of that, hospitals do whatever exchanges they can internally, hurting the overall welfare of the patients. So, the research was focused on finding certain mechanisms that give incentives to hospitals to reveal all their incompatible pairs and get an approximation of the optimal solution a lot better than the one given by the algorithm that disregards incentives.

There are generally two approaches to this problem: with or without distribution assumptions. Computer scientists are primarily interested in the latter. With this approach, we have two mechanisms: Mix-And-Match [13] and Weight-And-Match [22]. Weight-And-Match is a randomized mechanism with very good incentive properties and a 3/2 approximation ratio in the worst case. However, it works for 2 hospitals only. That is useful for ad-hoc arrangements between hospitals or small Kidney Exchange Programs. One possible future research direction would be to improve this mechanism to 4/3 approximation ratio, which is the proven lower bound with the specific incentive properties. Mix-And-Match is a randomized mechanism for many players that has very strong incentive properties but a bad approximation ratio of 2 in the worst case. However, simulations showed that in practice it is very close to optimal. According to the proven lower bounds, it is an open question whether one can create a mechanism for many players that has an approximation ratio of 8/7. Moreover, one can try to create mechanisms with very strong incentive properties, such as Group Strategyproofness, or Core. Another direction would be to devise a dynamic model for this problem with the incompatible pairs' pool to evolve over time.

Lastly, there is a new kind of exchange that can be used to increase efficiency: NEAD chains(Nonsimultaneous Extended Altruistic Donor chains) [18, 19]. The main problem with kidney exchange is that not very large exchanges can be conducted in practice due to the fact that all the transplantations and the nephrectomies in an exchange must be conducted at the same time. This is due to incentive reasons: if the intended patient of a donor already got his kidney, then he has no incentive to donate his own to the other patient. We cannot risk this situation occuring, because not only does the patient not get a kidney, he also loses his incompatible donor and thus cannot participate in a future exchange (Figure 8.1).

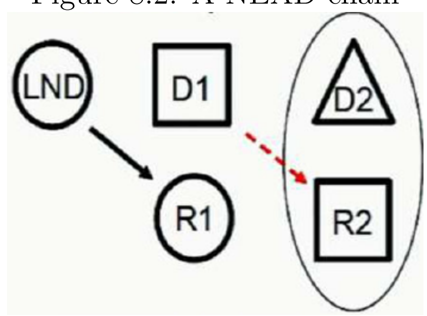
Figure 8.1: A conventional 2-way exchange



However, consider the scenario where an altruistic donor comes along who is willing to donate his kidney to a complete stranger. We can create a chain, as seen below

(Figure 8.2), that starts with the donor giving his kidney to a patient, and then that patient's incompatible donor gives to another patient and so forth. It is not so risky for these exchanges not to be done simultaneously, because in the case where a donor backs out, the patient expecting his kidney does not lose his incompatible donor, and therefore can participate in another exchange in the future. This solution was used in practice. For example, in the United States in 2011, Rick Ruzzamenti started a chain of this kind that has not been broken since and counted 30 transplantations by 2012 [64].

Figure 8.2: A NEAD chain



Bibliography

[1] Wolfe RA, Ashby VB, Milford EL, Ojo AO, Ettenger RE, Agodoa LYC, Held PJ, Port FK. Comparison of mortality in all patients on dialysis, patients on dialysis awaiting transplantation, and recipients of a first cadaveric transplant. In *N Engl J Med 341*,pages 1725-1730, 1999

[2] HRSA/OPTN Organ Procurement and Transplantation Network `http://optn.transplant.hrsa.gov/latestData/step2.asp`(accessed March 22, 2013)

[3] Jevons, William S. 1876. Money and the Mechanism of Exchange. *New York: D. Appleton and Company.* `http://www.econlib.org/library/YPDBooks/Jevons/jvnMME1.html`(accessed March 22, 2013).

[4] A. Gibbard, Manipulation of voting schemes, *Econometrica, vol. 41*, pp. 587-602, 1973.

[5] M. Satterthwaite, Strategyproofness and arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions, *Journal of Economic Theory, vol. 10*, pp. 187-217, 1975.

[6] W. Vickrey. Counterspeculation, auctions and competitive sealed tenders. *J. Finance*, 8-37, 1961.

[7] E.H. Clarke. Multipart pricing of public goods. *Public Choice*, 17-33, 1971.

[8] T. Groves. Incentives in teams. *Econometrica*, 617-631, 1973.

[9] John Geanakoplos. Three brief proofs of Arrow's impossibility theorem. *Econometrica* Economic Theory, 26:211-215, 2005.

[10] K. Arrow. Social Choice and Individual Values. *Yale University Press*, 1951

[11] A.Procaccia, M. Tennenholtz, Approximate Mechanism Design without Money,*EC 2009*, 177-186.

[12] D. Abraham, A. Blum, and T. Sandholm. Clearing algorithms for barter exchange markets: Enabling nationwide kidney exchanges. In *Proceedings of the 8th ACM Conference on Electronic Commerce*, pages 295-304, 2007

[13] I. Ashlagi, F. Fischer, I. Kash, and A.D. Procaccia. Mix and match. In *Proceedings of the 11th ACM Conference on Electronic Commerce*, pages 295-304, 2010.

[14] A.E. Roth, T.Sonmez, and M.U.Unver. Kidney exchange. *Quarterly Journal of Economics*,119:457-488, 2004.

[15] A.E. Roth, T. Sonmez, and M.U.Unver. Pairwise kidney exchange. *Journal of Economic Theory*, 125:151-188, 2005.

[16] A.E. Roth, T. Sonmez, and M.U.Unver. Efficient kidney exchange: Coincidence of wants in markets with compatibility-based preferences.*American Economic Review*, 97:828-851, 2007.

[17] A.E. Roth, T. Sonmez, and M. U. Unver. A kidney exchange clearinghouse in New England. *American Economic Review Papers and Proceedings*, 95(2):376-380, 2005a.

[18] M. A. Rees, J. E. Kopke, R. P. Pelletier, D. L. Segev, M. E. Rutter, A. J. Fabrega, J. Rogers, O. G. Pankewycz, J. Hiller, A. E. Roth, T. Sandholm, M. U. Unver, and R. A. Montgomery. A non-simultaneous extended altruistic donor chain. *New England Journal of Medecine*, 360:1096-1101, 2009.

[19] Ashlagi, Itai, David Gamarnik, Michael A. Rees and Alvin E. Roth, "The Need for (long) Chains in Kidney Exchange," *working paper*, May 2012.

[20] Ashlagi, Itai and Alvin E. Roth, "Individual rationality and participation in large scale, multi-hospital kidney exchange," *working paper*, January 2011

[21] P. Toulis and D. C. Parkes. A random graph model of kidney exchanges: Efficiency, individual-rationality and incentives. In *Proceedings of the 12th ACM Conference on Electronic Commerce*, pages 323-332, 2011.

[22] I. Caragiannis, A. Filos-Ratsikas, and A. D. Procaccia. An improved 2-agent kidney exchange mechanism. In *Proceedings of the 7th International Workshop on Internet and Network Economics*, 2011

[23] N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, editors. Algorithmic Game Theory. *Cambridge University Press*, 2007.

[24] Ross, Lainie Friedman, David T. Rubin, Mark Siegler, Michelle A. Josephson, J. Richard Thistlethwaite, Jr., and E. Steve Woodle, Ethics of a Paired-Kidney-Exchange Program,*The New England Journal of Medicine*, CCCXXXVI (1997), 1752-1755.

[25] Roth, A.E., "Incentive Compatibility in a Market with Indivisible Goods," *Economics Letters*, Vol. 9, 1982, 127-132.

[26] N. Nisan, A. Ronen, Algorithmic Mechanism Design, *STOC* 1999, 129-140.

[27] S. Dobzinski, N. Nisan, M. Schapira, Truthful Randomized Mechanisms for Combinatorial Auctions,*STOC* 2006, 644-652.

[28] S. Dobzinski, S. Dughmi, On the Power of Randomization in Algorithmic Mechanism Design, *FOCS* 2009, 505-514.

[29] S. Dughmi, T. Roughgarden, Q. Yan, From Convex Optimization to Randomized Mechanisms: Toward Optimal Combinatorial Auctions for Submodular Bidders, *STOC* 2011.

[30] A. Archer, E. Tardos, Truthful Mechanisms for One-Parameter Agents, *FOCS* 2001, 482-491.

[31] N. Andelman. Y. Azar, M. Sorani, Truthful Approximation Mechanisms for Scheduling Selfish Related Machines, *STACS* 2005, 69-82.

[32] P. Dhangwatnotai, S. Dobzinski, S. Dughmi, T. Roughgarden, Truthful Approximation Schemes for Single-Parameter Agents, FOCS 2008.

[33] K. Talwar, The Price of Truth: Frugality in Truthful Mechanisms, *STACS* 2003, 608-619.

[34] A. R. Karlin, D. Kempe, T. Tamir, Beyond VCG: Frugality of Truthful Mechanisms, *FOCS* 2005, 615-626.

[35] N. Chen, E. Elkind, N. Gravin, F. Petrov, Frugal Mechanism Design via Spectral Techniques, *FOCS* 2010, 755-764.

[36] D. Kempe, M. Salek, C. Moore, Frugal and Truthful Auctions for Vertex Covers, Flows, and Cuts, *FOCS* 2010, 745-754.

[37] Y. Singer, Budget Feasible Mechanisms, *FOCS* 2010, 765-774.

[38] N. Chen, N. Gravin, P. Lu, On the Approximability of Budget Feasible Mechanisms, *SODA* 2011, 685-699.

[39] O. Dekel, F. Fischer, A. Procaccia, Incentive Compatible Regression Learning, *SODA* 2008, 277-286.

[40] P. Lu, X. Sun, Y. Wang, Z. Zhu, Asymptotically Optimal Strategy-Proof Mechanisms for Two-Facility Games, *EC* 2010, 315-324.

[41] S. Dughmi, A. Ghosh, Truthful Assignment without Money, *EC* 2010, 325-334.

[42] N. Alon, F. Fischer, A. Procaccia, M. Tennenholtz, Sum of Us: Strategyproof Selection from the Selectors, 2011.

[43] N. Alon, M. Feldman, A. Procaccia, M. Tennenholtz, Strategyproof Approximation of the Minimax on Networks, *Mathematics of Operations Research, V.35(3)*, 513-526, 2010.

[44] D. Shmoys, E. Tardos, An Approximation Algorithm for the Generalized Assignment Problem, *Mathematical Programming, V.62(3)*, 461-474, 1993.

[45] L. Fleischer, M. Goemans, V. Mirrokni, M. Sviridenko, Tight Approximation Algorithms for Maximum General Assignment Problems, *SODA* 2006, 611-620.

[46] Ning Chen, Nick Gravin, Pinyan Lu: Mechanism Design without Money via Stable Matching. *CoRR abs/1104.2872*, 2011

[47] S. Barbera: "An Introduction to Strategy-proof Social Choice Functions" *Social Choice and Welfare 18(4)*, 619-653, 2001a.

[48] S. Barbera: "A Theorem on Preference Aggregation" *WP166 ,CREA-Barcelona Economics*, 2001b.

[49] Kobbi Nissim, Rann Smorodinsky, Moshe Tennenholtz: Approximately optimal mechanism design via differential privacy. *ITCS* 2012: 203-213

[50] K. Cechlarova, T. Fleiner and D. Manlove, The kidney exchange game, *Proc.SOR '05 , Eds. L. Zadnik-Stirn, S. Drobne,* 77-83, 2005

[51] K. Cechlarova and V. Lacko, The kidney exchange problem: How hard is it to find a donor?, *IM Preprint 4/2006*, 2006.

[52] P. Biro, K. Cechlarova. Inapproximability for the kidney exchange problem. *Information Processing Letters, 101(5)*, 199-202, 2007.

[53] J. Edmonds. Path, trees, and flowers. *Canadian Journal of Mathematics, 17:449-467*, 1965.

[54] P. Biro, D. F. Manlove, and R. Rizzi. Maximum weight cycle packing in directed graphs, with application to kidney exchange programs. *Discrete Mathematics, Algorithms and Applications, 1(4):499-517*, 2009.

[55] Zenios, Stefanos A. "Optimal Control of a Paired-Kidney Exchange Program." *Management Science, 48,* 328-342, 2002.

[56] Awasthi, P. and Tuomas Sandholm "Online Stochastic Optimization in the Large: Application to Kidney Exchange." In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2009.

[57] Unver, M. Utku "Dynamic Kidney Exchange." *Review of Economic Studies, 77 (1),*372-414, 2010.

[58] Yilmaz, Ozgur "Kidney Exchange: An Egalitarian Mechanism." *Journal of Economic Theory, 146(2),* 592-618, 2011

[59] Abdulkadiroglu, Atila, and Tayfon Sonmez, "House Allocation with Existing Tenants", *Journal of Economic Theory, LXXXVIII* , 233-260, 1999

[60] Roth, Alvin E., "Incentive Compatibility in a Market with Indivisibilities," *Economics Letters, IX* , 127-132, 1982.

[61] Shapley, Lloyd, and Herbert Scarf, "On Cores and Indivisibility," *Journal of Mathematical Economics*, 23-28, 1974

[62] Roth, Alvin E., and Andrew Postlewaite, "Weak versus Strong Domination in a Market with Indivisible Goods,"*Journal of Mathematical Economics, IV*, 131-137, 1977.

[63] Michel X. Goemans "Lecture Notes on matroid optimization",*Massachusetts Institude of Technology, Combinatorial Optimization*, 2009

[64] Kevin Sack, "60 Lives, 30 Kidneys, All Linked", *Article, NY Times*, 2012, `http://www.nytimes.com/2012/02/19/health/lives-forever-linked-through-kidney-transplant-chain-124.html?pagewanted=all&_r=0`(accessed March 22, 2013).

[65] S. Janson, T. Luczak, and R. Rucinski. Random Graphs. *A John Wiley & Sons*, 2000.