Mixed Matching Markets

BACHELOR THESIS

FernUniversität in Hagen,

Germany

Faculty of Mathematics and Computer Sciences

submitted by

David Schiess

from

Herisau (AR),

Switzerland

Matrikel-Nr.: 6788351

Prof. Dr. Winfried Hochstättler

August 2007

Preface

> "The effort of the economist is to see, to picture the interplay of economic elements. The more clearly cut these elements appear in his vision, the better; the more elements he can grasp and hold in his mind at once, the better. The economic world is a misty region. The first explorers used unaided vision. Mathematics is the lantern by which what before was dimly visible now looms up in firm, bold outlines. The old phantasmagoria disappear. We see better. We also see further."
>
> \- Irving Fisher

During my studies at the FernUniversität in Hagen I learned to appreciate the beauty and elegance of various concepts in mathematics. However, as an economic graduate I have sometimes not grasped the relation to reality. In this sense, I would like to express my sincere thanks to my academic advisor Prof. Dr. Winfried Hochstättler. He gave me the possibility to work on a mathematically interesting subject, which displays great applicability in economics.

I would like to express my gratitude to Robert Nickel for supporting me during all stages of the Bachelor thesis. The e-mail and telephone correspondence with him proved to be very inspiring, reassuring and valuable at the same time.

Moreover, many thanks go to Dr. Thomas Pfiffner for proof-reading large parts of the Bachelor thesis. I was lucky to have him as a fellow student and friend during my studies at the FernUniversität in Hagen.

Furthermore, I am deeply indebted to my parents Ernst and Renate as well as my brothers Raphael and Josua for their support during long years. I am proud of all of you. Finally, I owe very much to Claudia and her family. She has been very understanding during all stages of the Bachelor thesis and she has always motivated me and given me strength.

St. Gallen, August 2007 David Schiess

Contents

Abstract

The present Bachelor thesis deals with mixed matching markets. In the famous mixed matching market model of Eriksson and Karlander (2000) the characteristic of the two players involved in a contract determines the rigidity or flexibility of the contract. We give a further generalisation by introducing a model where the characteristic of the edges between any two players decides whether the contract is rigid or flexible. We therefore call it the decisive edges (DE) market model and show that it contains - among other models - the model of Eriksson and Karlander (2000) as a special case. Hochstättler et al. (2006) developed a polynomial auction algorithm for the mixed matching market model of Eriksson and Karlander (2000). We modify their algorithm to prove the existence of a stable outcome in our more general DE model. Finally, we show that our modified auction algorithm runs in $\mathcal{O}\left(n^4\right)$ time.

List of Figures

List of Tables

Notation

Uppercase Letters

A, $A^{(v;\mu)}$	set of arcs, set of arcs in the augmentation digraph $G^{(v;\mu)}$
C	positive constant
$D_i^{(v;\mu)}$	set of favourite workers of firm i
E	set of edges
F, F^*	set of flexible nodes, set of flexible edges
G, G_D, $G^{(v;\mu)}$	graph, digraph, augmentation digraph
M, M'	matchings
$M_{n\times n}$	set of $(n \times n)$-matrices
P, $\bar{P}$, P_U	set of firms, special set of firms, set of unmapped firms
Q, $\bar{Q}$, $Q_{2\mu}$	set of workers, special set of workers, set of doubly mapped workers
Q_R	set of rigidly mapped workers
Q_{2R}	set of workers with at least two rigid proposals
R, R^*	set of rigid nodes, set of rigid edges
S, S_P, S_Q	coalition, P-agents in the coalition S, Q-agents in the coalition S
V	set of nodes

Lowercase Letters

a, b	weight (productivity) matrices
e	edge
$f_{ij}^{(v;\mu)}$	benefit of firm i from a collaboration with worker j
g, h	real-valued functions
i, j	index for firms (P-agents), index for workers (Q-agents)
k, l	indices
m, n_0, r	natural numbers
n	number of firms, number of workers
p, q	firm, worker
s; y, z	start node; nodes
t	natural number or zero: $t = \min\{\lvert S_P\rvert, \lvert S_Q\rvert\}$
u, $\bar{u}$	payoff vector of firms, virtual payoff vector of firms
v	payoff vector of workers
x, x_{ij}	assignment matrix, primal variable

Uppercase Greek Letters

Δ, Δ_j	special quantities

Lowercase Greek Letters

α, α_{ij}	weight function, weight
$\lambda(S)$	worth of coalition S
μ	mapping

Calligraphic Letters

$\mathcal{O}(n^r)$	polynomial complexity time
$\mathcal{P}$	path

Abbreviations

BFS	breadth first search
DE	decisive edges
DLP	dual linear program
FB	flexibility bias
i.e.	in example
PLP	primal linear program
RB	rigidity bias
s.t.	such that
w.l.o.g.	without loss of generality
w.r.t.	with respect to

Chapter 1

Introduction

The theory of matching is an unusual blend of disciplines. Over the past years game theory, economics, computer science and combinatorial optimisation contributed to the theory. We will concentrate on two-sided matching in this Bachelor thesis. The probably most prominent example for two-sided matching is the labour market. Clearly, any agent in the labour market is either a firm or a worker, which explains why we call such a market **two-sided** or **bipartite**. Thus, we consider the finite and disjoint sets of firms P and workers Q in the labour market. We would like to mention that this is in contrast to product markets, where the same agent can act as a seller and as a buyer.[1] Obviously, the sellers and buyers are not two disjoint subsets of the set of all agents in product markets. Hence, product markets are not two-sided. Finally, we use the term **one-to-one matching** or simply **matching**, since we will study the setting where a firm hires exactly one worker and where all workers will work for one firm only.[2]

There are two fundamental models for two-sided matching markets: The **marriage model** of Gale and Shapley (1962) and the **assignment game** of Shapley and Shubik (1972). In the marriage model, we consider the set of men and women who are eligible to marriage in some small village. Each individual has a preference list of his/her[3] acceptable partners. The problem is to find a marriage s.t. there is no pair (i, j), where both, man i and woman j, are not matched but prefer each other over their current partners. Such a marriage will be called **stable**. Gale and Shapley (1962) proved the existence of a stable marriage with their famous "men-propose-women-dispose" algorithm in the case where preference lists are strict.

In the assignment game, we attach a money value to each edge and call it the edge's **weight**. Clearly, money is a continuous variable and hence, the market will be continuous. In the labour market, this money value attached to any edge connecting a firm i with a worker j can be thought of as the total productivity that installs if worker i is employed by firm j. This productivity can be freely transferred between the agents.[4] The possibility of monetary transfers makes the assignment game **flexible** compared to the **rigid** marriage problem. A solution of an assignment

[1] Imagine an agent who acts as a seller if the price is sufficiently high. For low enough prices the same agent is likely to act as a buyer.

[2] For a highly educated worker applying for a high position in a firm such a setting is very realistic. For example, we could study the matching of firms and chief financial officers. On the other hand, if we study low level jobs, then we should allow that a firm hires several workers for the same job instead.

[3] For the sake of brevity we omit any female forms if possible in the remainder of the thesis.

[4] In the current example of the labour market, the productivity can be transferred through the wage the worker receives.

game consists of a matching of firms with workers[5] and an allocation of the corresponding weight and will be called an **outcome**. If no pair receives less than the weight of its connecting edge, then we call the solution a **stable outcome**. Technically speaking, the assignment game amounts to the determination of a maximally weighted matching in the **bipartite graph** of firms and workers. Shapley and Shubik (1972) showed the existence of stable outcomes via duality arguments of linear programming. The classic algorithm for weighted bipartite graphs is undisputably Kuhn's Hungarian method[6].

The practical relevance of both models is immense. For instance, let us consider some new graduates in mathematics. They have the discrete choice of entering either the public or the private labour market. To be more specific, they can become a teacher of mathematics at some high school or they can work in the risk department of some insurance company. If a graduate decides to become a teacher, then his salary will be fixed. This part of the market is appropriately described with the marriage model. On the other hand, if a graduate chooses to work for an insurance company, then his salary will no longer be fixed. Instead, the graduate and the company will contract on the salary among other job characteristics. This part of the market can be appropriately modelled with the assignment game, which allows monetary transfers. Taken together, it is clearly tempting to study the entire market - the private and the public labour market - simultaneously within a single model. We will refer to such models, which contain rigid as well as flexible aspects, as **mixed matching markets**.

We would like to give a historical remark that highlights the practical relevance of matching theory.[7] The American Hospital Association and the Association of American Medical Colleges agreed in 1951 to use a central algorithm to match medical students with medical interns of hospitals. For a liberal country like the United States, this seems to be quite remarkable. The agreement to use a central matching was caused by a disastrous market situation. The number of positions for interns was greater than the number of medical students applying for such positions. The resulting considerable competition among hospitals manifested itself in the fact that hospitals attempted to finalise binding agreements with student earlier than their principal competitors. This led to a costly and inefficient market situation: Hospitals did not know the final grades of their appointed students while the students and the medical schools found that schooling was disrupted by the tedious process of seeking desirable interns. The central matching algorithm[8] was implemented as a voluntary procedure. Students and hospitals were free to arrange their matches outside of the system. Despite this voluntariness, the participation rates initially exceeded 95 percent.

The marriage problem and the assignment game lead to very similar results: Equality of the core and the set of stable outcomes as well as the lattice structure of the core. Thus, it is not very surprising that Roth and Sotomayor (1996) asked for an explanation of these similarities in the two mentioned models. Eriksson and Karlander (2000) addressed the challenge of Roth and Sotomayor (1996) by presenting a mixed matching market model[9] that contains the marriage

[5]To be more general: A matching of P-agents with Q-agents.

[6]See the origin work of Kuhn (1955) or the more recent treatment of Frank (2004).

[7]See Roth (1984), Roth (1991) or the comprehensive treatment of two-sided matching of Roth and Sotomayor (1999) for a thorough examination.

[8]It was called NIMP: National Intern Matching Program.

[9]The so-called RiFle (RigidFlexible) assignment game.

problem and the assignment game as special cases. They allow for rigid and flexible players in their model and define an edge to be rigid if at least one of the players involved is rigid. Eriksson and Karlander (2000) finally provided a pseudo-polynomial auction algorithm to prove the existence of stable outcomes in their mixed matching market model. Based on the ideas of Eriksson and Karlander (2000), Hochstättler et al. (2006) constructed a polynomial auction algorithm and proved that it runs in $\mathcal{O}\left(n^4\right)$ time where $2n$ denotes the number of players. Their auction algorithm will serve as a benchmark for our work. Parallel to this branch of the literature, Sotomayor (2000) also showed that there is always a stable outcome in a mixed matching market model which is very similar to the model of Eriksson and Karlander (2000). Finally, Hochstättler et al. (2005) derived another polynomial algorithm from the key lemmas of Sotomayor (2000) and showed that this algorithm also runs in $\mathcal{O}\left(n^4\right)$ time. In this thesis we give a further generalisation of the mixed matching market model of Eriksson and Karlander (2000). In contrast to the model of Eriksson and Karlander (2000), we define the rigidity or flexibility of any edge to be independent of the players involved. We call the resulting market the **decisive edges (DE)** market model and show the usefulness of such a generalisation. We introduce a modification of the auction algorithm of Hochstättler et al. (2006) and exploit it to prove the existence of a stable outcome in our general DE model. Furthermore, we show that our modified auction algorithm runs in $\mathcal{O}\left(n^4\right)$ time, too. Lastly, we would like to mention the interesting matching market model of Fujishige and Tamura (2004). They generalise the mixed matching market of Eriksson and Karlander (2000) by modelling the preferences of agents on each side over the agents on the other side with concave utility functions. However, our model is not contained in theirs and vice versa.

The thesis is structured as follows. First, we present some basic mathematical concepts that will help the reader to understand the remainder of the thesis. Chapter 3 then presents different matching market models. We begin with two famous special cases: The assignment game and the marriage model. These two models are contained in the mixed matching market model of Eriksson and Karlander (2000). Afterwards, we give a further useful generalisation: The decisive edges (DE) market model. We will show that the model of Eriksson and Karlander (2000) and many other models are simply special cases of this DE model, which will be studied in the remainder of the thesis. In chapter 4 we introduce a modification of the auction algorithm of Hochstättler et al. (2006). We then perform the main task of the thesis: We exploit the mentioned modified auction algorithm to give a constructive proof of the existence of a stable outcome in our general DE market model. Besides the correctness of the modified auction algorithm, we establish the result that it runs in $\mathcal{O}\left(n^4\right)$ time. Lastly, we close the thesis by giving some concrete examples and a comparison to the auction algorithm of Hochstättler et al. (2006).

Chapter 2

Mathematical Preliminaries

In this chapter we introduce the mathematical concepts that will be exploited in the remainder of the thesis. For the sake of brevity, we will customise these preliminaries to our future needs. For a more general and more thorough treatment we refer the interested reader to Ahuja et al. (1993), Hochstättler (1999), Jungnickel (2005) and Aigner (2006).

2.1 Some Concepts in Graph Theory

We first give the definition of a digraph and a graph, respectively.

Definition: 2.1 (Digraph and Graph) Let V and A denote a finite set of nodes (vertices) and a finite set of arcs, respectively. We denote an arc by the ordered pair (z_1, z_2) with $z_1, z_2 \in V$ where z_1 and z_2 represent the head and tail node, respectively. We call the tuple $G_D = (V, A)$ a **directed graph** or simply a **digraph**.

If the orientation of the arcs is irrelevant, then we call them edges and denote the set of all edges with E. We denote an edge by the unordered pair (z_1, z_2) with $z_1, z_2 \in V$ where z_1 and z_2 represent the two end nodes of the edge.[1] Finally, we call the tuple $G = (V, E)$ a **graph**. $\square$

With a slight abuse of notation we adopt the convention that the first entry z_1 of an edge (z_1, z_2) denotes a firm while the second entry z_2 represents a worker. We will need the following definitions to describe a matching.

Definition: 2.2 (Incident Nodes and Edges) We consider the graph $G = (V, E)$. A node $z \in V$ is called **incident** to an edge $e = (z_1, z_2) \in E$ if $z = z_1$ or $z = z_2$. Finally, two edges $e_1 = (y_1, z_1), e_2 = (y_2, z_2) \in E$ are **incident** if $\{y_1, z_1\} \cap \{y_2, z_2\} \neq \emptyset$. $\square$

Lastly, the concept of a path will be fundamental in our modified auction algorithm 4.1.

Definition: 2.3 (Path in a Digraph) A **path** $\mathcal{P} = (z_1, z_2, ..., z_l)$ in a digraph $G_D = (V, A)$ is a sequence of vertices $z_i \in V$ s.t. $(z_i, z_{i+1}) \in A$ and $z_i \neq z_j$ for all $i \neq j$. We say that the node z_l is reachable from the node z_1 with the path $\mathcal{P}$. $\square$

In the remainder of the thesis, we will rule out loops and multigraphs. Instead, we will study labour markets, where an agent is either a firm or a worker. Moreover, we want to ensure that

[1] A loop is an edge connecting a node to itself. Thus, we can have $z_1 = z_2$ in case of a loop.

each firm can contract with each worker and vice versa. This leads to the following special case of a graph.

Definition: 2.4 (Complete Bipartite Graph) A graph $G = (V, E)$ is called **bipartite** if it allows for a partition of the set of nodes $V = P \dot{\cup} Q$ s.t. each edge has one end node in P and the other in Q. A bipartite graph with $|P| = n$ and $|Q| = m$ is called **complete** if it satisfies $E = \{(p, q) \mid p \in P,\ q \in Q\}$, which implies that we have $|E| = nm$. □

Note that a complete bipartite graph contains almost all the relevant aspects of the labour market we want to study. We can define P as the set of firms and Q as the set of workers. Clearly, these two sets are finite and disjoint in the labour market.

2.2 Matching in Complete Weighted Bipartite Graphs

As previously mentioned, we only have edges connecting a firm with a worker in complete bipartite graphs. We now attach a money value to each edge (i, j) that can be thought of as the total productivity that installs if a worker j is employed by a firm i. Obviously, such a productivity must be nonnegative.

Definition: 2.5 (Weighted Graph) A graph $G = (V, E, \alpha)$ is called **weighted** if it has a weight function $\alpha : E \to \mathbb{R}_+$. □

From now on we assume a complete weighted bipartite graph $G = (V, E, \alpha)$ with $|P| = |Q| = n$. This assumption is innocious, since we can always introduce dummy nodes with zero weights. We next turn to the fundamental definition of a matching. Because we are interested in the situation where each firm wants to hire one worker and each worker can only be employed by one firm, the matching will be one-to-one.[2] For the sake of brevity, we simply use the term matching.

Definition: 2.6 (Matching in a Weighted Bipartite Graph, Weight of a Matching) We consider a weighted bipartite graph $G = (V, E, \alpha)$ with $V = P \dot{\cup} Q$. A **matching** $M \subset E$ is a set of pairwise non-incident edges. We call two nodes $p \in P$ and $q \in Q$ **matched with each other** if $(p, q) \in M$. Moreover, we call a node **unmatched** if it is not incident with an edge of M. We define the weight of a matching M as $\alpha(M) = \sum_{e \in M} \alpha(e)$. Finally, we call M a **maximally weighted matching** if $\alpha(M) \geq \alpha(M')$ for all other matchings M'. □

We will see that the assignment game boils down to finding a maximally weighted matching in a complete weighted bipartite graph.

[2] The comprehensive treatment of two-sided matching of Roth and Sotomayor (1999) also contains a chapter on many-to-one matching. Additionally, the interested reader can also find examples of non-two-sided matching such as the roomate problem or the man-woman-child problem. In all the mentioned problems however, we do not necessarily have stable outcomes.

2.3 Algorithms and Complexity

Let us give the following rather informal definition of an algorithm.

DEFINITION: 2.7 (ALGORITHM) We define an **algorithm** as an exact finite description of a sequence of a finite number of steps that establish a certain goal. However, each step must be unique and efficiently executable. Finally, the algorithm must terminate after the finite number of steps. □

We will formulate our modified auction algorithm in pseudo-code. Moreover, we will sometimes violate the uniqueness condition for the sake of expositional ease. However, all such violations can easily be remedied by certain rules like selecting the node with the lowest index in case there are several nodes to operate on at some step in the algorithm.

Besides the correctness, a major characteristic of any algorithm is its runtime behavior. We will make use of the $\mathcal{O}$-notation to give upper bounds on the complexity of our modified auction algorithm and to compare its runtime behavior with those of other algorithms.

DEFINITION: 2.8 (COMPLEXITY, $\mathcal{O}$-NOTATION) We consider the functions $g : \mathbb{N} \to \mathbb{N}$ and $h : \mathbb{N} \to \mathbb{N}$. We write $g = \mathcal{O}(h)$ if there exist $C > 0$ and $n_0 \in \mathbb{N}$ s.t. $g(n) \leq Ch(n)$ for all $n \geq n_0$. □

Note that we have to read the equation $g = \mathcal{O}(h)$ from the left to the right.[3] Moreover, we would like to mention that if a problem can only be solved with algorithms displaying exponential growth in their runtimes, then the problem is generally considered to be insolvable. Thus, algorithms of interest exhibit at most polynomial runtimes[4]. Fortunately, we will be able to prove that our modified auction algorithm runs in $\mathcal{O}(n^4)$ time.

2.4 Breadth First Search

Our modified auction algorithm will exploit the breadth first search as a subroutine. We already present this standard search method here, since we want to focus on the really relevant aspects of the algorithm later.

The breadth first search (BFS) determines all nodes that are reachable in a given digraph from a given start node. For this purpose, BFS first visits all nodes that are reachable with only one arc from the start node, marks these nodes as visited and stores them in a queue. As soon as all such directly reachable nodes are visited, we remove a node from the queue (current node), visit all unvisited nodes that are directly reachable from the current node and put them on the queue. BFS continues in this manner until some termination condition is met or the queue has become empty. The following algorithm in pseudo-code implements the breadth first search. It must be called with BFS((G_D, s)) where G_D is some directed graph and s denotes the start node of the search.[5]

[3] Otherwise, we could get the contradiction $n = n^2$ from $n = \mathcal{O}(n^2)$ and $n^2 = \mathcal{O}(n^2)$.

[4] That is, their runtime functions must be in $\mathcal{O}(n^r)$ for some $r \in \mathbb{N}$.

[5] Our modified auction algorithm will exploit BFS to determine all nodes that are reachable in the augmentation digraph from the set of doubly mapped workers. Hence, we use some variant of the presented algorithm. Specifically, we execute lines 3 and 4 for all doubly mapped workers. Moreover, we will not use a termination condition and read off the reached nodes from the vector *Predecessor* in the very end.

Algorithm 2.9: Breadth First Search (BFS)

```
Queue ← ∅
Predecessor ← 0
Predecessor[s] ← −1                        ◇ The path begins here
Queue.append(s)                            ◇ Put s on the queue
while Queue ≠ ∅ do
    y = Queue.top()                        ◇ Take y and remove it from the queue
    for z ∈ y.Neighbourhood do             ◇ There is an arc from y to z
        if Predecessor[z] = 0 then
            Predecessor[z] ← y
            if z satisfies termination condition then
                P ← (s, ..., z)
            else
                Queue.append(z)
            end if
        end if
    end for
end while
```

The next chapter will present different mixed matching market models.

Chapter 3

Matching Market Models

In this chapter we first present the assignment game of Shapley and Shubik (1972) and afterwards the marriage model of Gale and Shapley (1962). We use general notation from the very beginning, since these two models are special cases of the quite general mixed matching market model of Eriksson and Karlander (2000). Finally, we introduce a further useful generalisation: The decisive edges (DE) market model. This most general model will be exploited in the remainder of the thesis. The entire chapter is devoted to develop some intuition for the presented matching market models. We therefore do not attempt to give a thorough introduction to the various existing models.[1]

As previously mentioned, we always assume w.l.o.g. that $|P| = |Q| = n$. The P- and the Q-agents will also be called the firms and the workers of a labour market, respectively. Each firm can contract with each worker and vice versa. However, each firm is interested in hiring exactly one worker and each worker can only be employed by one firm. Hence, the goal is to find stable one-to-one matchings[2]. For the sake of brevity, we simply use the term matching from now on. Instead of firms and workers, the reader can imagine sellers and buyers in a market where each seller possesses one indivisible good and each buyer is interested in purchasing one such good. The sets of sellers and buyers are finite, disjoint and denoted with P and Q, respectively. We denote a nonnegative real number (weight) α_{ij} with each partnership (p_i, q_j) with $i, j \in \mathbb{N}_n$. This number can be thought of as the difference between the reservation price of the buyer and the seller. Let us come back to our guiding example: The labour market. We then interpret α_{ij} as the worth of productivity when the worker q_j is hired by the firm p_i. For notational ease, let i and j be the index for firms and workers for the remainder of the thesis.

Besides $V = P\dot{\cup}Q$, we now introduce the additional partition of players $V = R\dot{\cup}F$ where R and F denote the set of rigid and flexible players, respectively. Rigid agents want a fixed salary while flexible agents prefer to contract on the salary. Moreover, we replace the weight function $\alpha : E \to \mathbb{R}_+$ with the two **productivity matrices** $a, b \in M_{n\times n}(\mathbb{R}_+)$ in the sense that[3] $\alpha(i,j) = \alpha_{ij} = a_{ij} + b_{ij}$ for each edge $(i,j) \in E$. Additionally, we define a **payoff** as the

[1] The interested reader is referred to Roth and Sotomayor (1999), Eriksson and Karlander (2000) and Jin (2005).

[2] The comprehensive treatment of two-sided matching of Roth and Sotomayor (1999) contains a chapter on many-to-one matching.

[3] As previously mentioned, we use general notation already here. The meaning of this notation will be explained in section 3.3.

pair (u, v) of the vectors $u, v \in \mathbb{R}^n$. The vectors u and v represent the benefit of the firms and workers, respectively.

3.1 Special Case I: Assignment Game

If we set $R = \emptyset$ in the mixed matching market model of Eriksson and Karlander (2000)[4], then we obtain the famous assignment game of Shapley and Shubik (1972). Money plays an important role in this cooperative game. The worth of any coalition of players S is determined solely with the best pairwise combination that the members of the coalition can form. Thus, we have

1. $\lambda(S) = 0$ if S contains either only P-agents or only Q-agents,
2. $\lambda(S) = a_{ij} + b_{ij}$ if $S = (i, j)$ with $i \in P$ and $j \in Q$,
3. $\lambda(S) = \max\{\lambda(i_1, j_1) + \lambda(i_2, j_2) + ... + \lambda(i_t, j_t)\}$ with $\{(i_1, j_1), (i_2, j_2), ..., (i_t, j_t)\} \subset S_P \times S_Q$ where S_P and S_Q denote the P- and Q-agents in S and where $t = \min\{|S_P|, |S_Q|\}$.

The rules of the game allow the members of a coalition to split their worth in any way they like. Hence, we do not only allow that monetary transfers are made between matched partners[5] but we do also allow for transfers between unmatched members of a coalition.[6] Clearly, we must have $\sum_{i \in S_P} u_i + \sum_{j \in S} v_j = \lambda(S)$. The problem is to determine $\lambda(S)$ for the given productivity matrices a and b and is called the assignment problem. Of course, we are especially interested in computing $\lambda(P \dot{\cup} Q)$, since this is the maximum total payoff available to the players of the game.

The assignment problem is equivalent to the problem of finding a maximally weighted matching in the complete weighted bipartite graph $G = (V, E, a, b)$ with the productivity matrices a and b. Fortunately, Kuhn (1955) developed the popular Hungarian method[7], which finds a maximally weighted matching in weighted bipartite graphs. The Hungarian method clearly influenced our work, as the reader will see when we present our modified auction algorithm in section 4.2 as well as later in example 4.8 where we solve an instance of the assignment game. We can cast the assignment problem into the following linear program (PLP).

Problem 3.1 (Primal Linear Program) *Maximise* $\sum_{i,j}(a_{ij} + b_{ij}) \cdot x_{ij}$ *subject to* $\sum_i x_{ij} \leq 1$ *for all* $j \in \mathbb{N}_n$, $\sum_j x_{ij} \leq 1$ *for all* $i \in \mathbb{N}_n$ *and* $x_{ij} \geq 0$ *for all* $i, j \in \mathbb{N}_n$.

If $x_{ij} = 1$, then i and j form a partnership and $x_{ij} = 0$ otherwise. Clearly, $\sum_i x_{ij} = 0$ means that i is unassigned.

We next give the corresponding dual linear program (DLP).

Problem 3.2 (Dual Linear Program) *Minimise* $\sum_{i \in P} u_i + \sum_{j \in Q} v_j$ *subject to* $u_i \geq 0$, $v_j \geq 0$ *and* $u_i + v_j \geq a_{ij} + b_{ij}$ *for all* $i, j \in \mathbb{N}_n$.

[4] See section 3.3.

[5] These are the obvious transfers, since a firm will pay the employed worker a certain wage.

[6] This assumption implies that there can also be monetary transfers between workers, for instance in a labour union, and between firms, for instance in an employer association.

[7] See the origin work of Kuhn (1955) or the more recent treatment of Frank (2004).

It can be shown that there exists an integer solution to the primal linear program.[8] Hence, we can conclude that the above dual linear program must have an optimal solution. According to the fundamental duality theorem of Dantzig (1963, p. 129) we therefore have identical values for the objective functions of the PLP and the DLP, respectively. Thus, if the matrix x is an optimal assignment and if (u, v) is a solution to the DLP, then we get

$$\sum_{i \in P} u_i + \sum_{j \in Q} v_j = \sum_{i,j} (a_{ij} + b_{ij}) \cdot x_{ij} = \lambda (P \dot{\cup} Q) . \tag{3.1}$$

We highlight the fact that the pair (u, v) of the dual variables u and v corresponds to the payoff of the game. Let us now think about the solution of the game, which we call an **outcome**. According to (3.1) an outcome consists of a matching and the payoff (u, v). We next define the stability of the payoff and the outcome.

DEFINITION: 3.3 (PAYOFF AND OUTCOME STABILITY) A **payoff** (u, v) is called **stable** if

(i) $u_i + v_j \geq a_{ij} + b_{ij}$ for all edges $(i, j) \in P \times Q$.

A **stable outcome** $(u, v; \mu)$ consists of a stable payoff (u, v) and a bijective map $\mu : P \to Q$ s.t.

(ii) $u_i \geq 0$ and $v_j \geq 0$ for all $(i, j) \in P \times Q$,

(iii) $u_i + v_j = a_{ij} + b_{ij}$ for all $(i, j) \in P \times Q$ if $j = \mu (i)$. □

Let us assume for the moment that we have $u_i + v_j < a_{ij} + b_{ij}$ for some edge $(i, j) \in P \times Q$. Obviously, firm i and worker j can earn more if they leave their current partners and collaborate with each other. Such a situation cannot be stable and hence, the pair (i, j) is called a **blocking pair**. We note that condition (i) in the above definition 3.3 prevents any blocking pairs, while condition (ii) ensures individual rationality.

This is the way how Shapley and Shubik (1972) showed that the stable payoffs of the assignment game (u, v) exist and that they are the solution of a dual linear program to the primal linear program for maximally weighted bipartite matchings. Moreover, they proved the equality of the core[9] and the set of stable outcomes as well as the lattice structure of the core. Interestingly, we will see the same results in the following marriage model.

3.2 Special Case II: Marriage Model

If we set $F = \emptyset$ in the mixed matching market model of Eriksson and Karlander (2000)[10], then we obtain the famous marriage model of Gale and Shapley (1962). We consider all men and women eligible for marriage in some small and isolated village. Thus, we imagine the P-agents to be the male and the Q-agents to be the female marriage candidates. Obviously, the two sets P and Q are then finite and disjoint as postulated. Each man has a preference list[11] over all

[8] The interest reader is referred to Dantzig (1963, p. 318).

[9] The core of a game is the set of undominated outcomes. Since the set of stable outcomes in the assignment game is defined w.r.t. all kinds of coalitions, it trivially coincides with the core.

[10] See section 3.3.

[11] For the agents' preferences Gale and Shapley (1962) imposed the completeness, the transitivity and the indepence assumption. All mentioned assumptions are standard in economics.

women and each woman has a preference list over all men. We represent these preference lists with the matrices a and b. For example, $a_{ik} > a_{il}$ then means that man i strictly prefers woman k to woman l. On the other hand, woman j prefers man k to man l if $b_{kj} \geq b_{lj}$ whereas she is indifferent between the two men if $b_{kj} = b_{lj}$. If a man and a woman both consent to marry one another, then they may proceed to do so. The problem is to find a set of marriages s.t. there exists no pair (i, j) that prefers each other over their current partners. We next define the stability of the payoff and the outcome.

Definition: 3.4 (Payoff and Outcome Stability) A **payoff** (u, v) is called **stable** if

(i) $u_i \geq a_{ij}$ or $v_j \geq b_{ij}$ for all $(i, j) \in P \times Q$.

A **stable outcome** $(u, v; \mu)$ consists of a stable payoff (u, v) and a bijective map $\mu : P \to Q$ s.t.

(ii) $u_i \geq 0$ and $v_j \geq 0$ for all $(i, j) \in P \times Q$,

(iii) $u_i = a_{ij}$ and $v_j = b_{ij}$ for all $(i, j) \in P \times Q$ if $j = \mu(i)$. □

Let us assume for the moment that we have $u_i < a_{ij}$ and $v_j < b_{ij}$ for some edge $(i, j) \in P \times Q$. Obviously, man i and woman j are then better off if they leave their current partners and marry each other. Such a situation cannot be stable and hence, the pair (i, j) is called a **blocking pair**. Note that condition (i) in the above definition 3.4 prevents any blocking pairs, while condition (ii) ensures individual rationality. Moreover, we note that the outcome is completely determined by the matching, since the matching implies the payoff as can be seen from condition (iii) in the above definition 3.4.

Gale and Shapley (1962) proved with their famous "men-propose-women-dispose" algorithm[12] that there is always a stable outcome when preferences are strict. We note that the procedure PlaceRigidProposals of our modified auction algorithm in section 4.2 is based on the "men-propose-women-dispose" algorithm. We will see this later in example 4.7 where we solve an instance of the marriage problem. Finally, Gale and Shapley (1962) proved the equality of the core[13] and the set of stable outcomes as well as the lattice structure of the core. Note that we have encountered the same results in the assignment game, too. Thus, it is not very surprising that Roth and Sotomayor (1996) asked for an explanation of these similarities in the marriage model and the assignment game. Eriksson and Karlander (2000) addressed the challenge of Roth and Sotomayor (1996) by giving a mixed matching market model that contains the marriage problem and the assignment game as special cases. The next section deals with their model.

[12] Roth and Vate (1990) show that there is an alternative to the "men-propose-women-dispose" algorithm. They start with any matching and randomly select any blocking pair to derive a new matching. Roth and Vate (1990) prove that such a random sequence of matchings converges to a stable matching. Thus, they provide a family of alternative algorithms to reach a stable matching.

[13] Again, the core of a game is the set of undominated outcomes. The difference between the set of stable outcomes and the core in the marriage model is that the core is undominated w.r.t. all coalitions whereas the set of stable outcomes is defined w.r.t. certain kinds of coalitions only: Single coalitions and pairs of a man and a woman.

3.3 Mixed Matching Model of Eriksson and Karlander (2000)

In contrast to the two previous models, Eriksson and Karlander (2000) simultaneously allow for rigid and flexible players. They then define the set of rigid and flexible edges $R^* = \{(i,j) \in P \times Q \mid i \in R \text{ or } j \in R\}$ and $F^* = \{(i,j) \in P \times Q \mid i \in F \text{ and } j \in F\}$, respectively. Thus, they assume that a contract will be fixed if at least one of the two parties prefers a fixed salary. Obviously, this is an arbitrary assumption. Note that we get $F^* \cup R^* = P \times Q$. If worker j gets employed by firm i and if $(i,j) \in R^*$, then we set $u_i = a_{ij}$ and $v_i = b_{ij}$. On the other hand, if $(i,j) \in F^*$ and firm i hires worker j, then we set $u_i + v_j = a_{ij} + b_{ij}$. Thus, we can interpret v_j as the worker's salary. This salary is fixed if the contract is rigid. If the contract is flexible, then the worker's wage is no longer fixed but it must be negotiated. Consequently, we only postulate that the sum of the firm's and the worker's benefit equals the total productivity from the collaboration in this case. Taken together, we consider the complete weighted bipartite graph $G = (V, E, a, b)$ with $V = P \dot{\cup} Q = R \dot{\cup} F$ and the two nonnegative productivity matrices a and b. We next give the stability definition in the mixed matching market model of Eriksson and Karlander (2000).

Definition: 3.5 (Payoff and Outcome Stability) A **payoff** (u, v) is called **stable** if the following two conditions are satisfied for every edge $(i,j) \in P \times Q$:

(i) $u_i + v_j \geq a_{ij} + b_{ij}$ if $(i,j) \in F^*$,

(ii) $u_i \geq a_{ij}$ or $v_j \geq b_{ij}$ if $(i,j) \in R^*$.

A **stable outcome** $(u, v; \mu)$ consists of a stable payoff (u, v) and a bijective map $\mu : P \to Q$ s.t.

(iii) $u_i \geq 0$ and $v_j \geq 0$ for all $(i,j) \in P \times Q$,

(iv) $u_i + v_j = a_{ij} + b_{ij}$ if $j = \mu(i)$ and $(i,j) \in F^*$,

(v) $u_i = a_{ij}$ and $v_j = b_{ij}$ if $j = \mu(i)$ and $(i,j) \in R^*$. □

Note that conditions (i) and (ii) in the above definition 3.5 prevent any blocking pairs, while condition (iii) ensures individual rationality. Finally, conditions (iv) and (v) set the payoffs of flexible and rigid matches. We would like to highlight the fact that the above stability definition 3.5 specialises to the stability definition 3.3 of the assignment game if we set $R = \emptyset$ and to the stability definition 3.4 of the marriage model in the special case of $F = \emptyset$. This should not come as a surprise, since the assignment game and the marriage model are special cases of the current mixed matching market model of Eriksson and Karlander (2000).

Finally, Eriksson and Karlander (2000) proved with a pseudo-polynomial auction algorithm that there always exists a stable outcome in the presented mixed matching market model. Based on their work, Hochstättler et al. (2006) developed a polynomial auction algorithm that runs in $\mathcal{O}(n^4)$ time. We will give a modification of their auction algorithm in section 4.2.

3.4 Decisive Edges Market Model

We now give a useful generalisation of the previous mixed matching market model of Eriksson and Karlander (2000). In contrast to the model of Eriksson and Karlander (2000), the nature of any contract will not depend on the characteristics of the players involved, but it will be determined solely by the edge that connects any two players. We recall that a contract is flexible in the model of Eriksson and Karlander (2000) if and only if both players are flexible. Put differently, a contract is rigid if at least one of the players is rigid. Thus, the set of rigid and flexible edges R^* and F^*, respectively, are defined as

$$R^* = \{(i,j) \in P \times Q \mid i \in R \text{ or } j \in R\} \tag{3.2}$$

and

$$F^* = \{(i,j) \in P \times Q \mid i \in F \text{ and } j \in F\} \,. \tag{3.3}$$

Let us therefore refer to the model of Eriksson and Karlander (2000) as the **rigidity bias (RB) market model**.

Alternatively, we could study a market where a contract is rigid if and only if both players are rigid. Put differently, a contract is flexible if at least one of the players is flexible. Hence, we could switch the logical operators in the definitions of R^* and F^* of the RB model in (3.2) and (3.3), respectively. The resulting market model is consequently called the **flexibility bias (FB) market model**. We then have

$$R^* = \{(i,j) \in P \times Q \mid i \in R \text{ and } j \in R\} \tag{3.4}$$

and

$$F^* = \{(i,j) \in P \times Q \mid i \in F \text{ or } j \in F\} \,. \tag{3.5}$$

Regardless of whether we consider the RB or the FB market model, we clearly have strict rules that determine the set of rigid and flexible edges R^* and F^*, respectively, by exploiting the nature of the players involved. We could make up different rules for these sets and end up with various different mixed matching market models. However, all these **"rule-based" market models** are trivially special cases of the market model that uses no rule at all: The **decisive edges (DE) market model**. In the DE model we do not model rigid and flexible players. Instead, we directly define whether an edge is rigid or flexible. Moreover, we do so in an arbitrary general way. This means that we can define the rigidity or flexibility of each edge individually, i.e. completely independent of all other edges. Put differently, the sets of rigid and flexible edges R^* and F^*, respectively, are allowed to form an arbitrary partition of $P \times Q$ in the DE model. However, the definition of R^* and F^* is the only difference to the RB model, which means that definition 3.5 of a stable payoff and outcome also applies to the current DE model.[14] We will fruitfully exploit the generality of the DE market model in the remainder of the thesis: We will prove the existence of stable outcomes in the DE model. Clearly, all the results that will be developed for the DE model carry over to the special cases, i.e. the RB and the FB market model.

[14]For the same reason, definition 3.5 of course applies to the FB model and any other "rule-based" market model, too.

Before we move on to solve the DE model, we want to think about the rigidity and flexibility of contracts and thereby show that the DE market model represents a useful generalisation of the mixed matching market model of Eriksson and Karlander (2000). In the labour market, we mainly distinguish between private firms and public organisations. Intuitively, private firms tend to behave more flexibly compared to public organisations. On the other side of the labour market, we mainly discern between members and non-member of labour unions[15]. Labour unions give wage recommendations and some agents feel obliged to follow them while others do not.[16] Moreover, whether an agent feels more or less obliged to follow such recommendations probably depends on the matching partner. If a private firm hires for instance a prominent member or even a leader of a labour union, then it is very likely to follow the recommendations. In a "rule-based" market model we can account for this situation by defining the private firm as a flexible and the prominent labour unionist as a rigid player, respectively, and by defining an edge as rigid if at least one of the players involved is rigid (i.e. adopting the RB model). If the same private firm (modelled as a flexible player) contracts with a completely unknown labour unionist (clearly a rigid player), then it is likely to contract on the salary instead of following a fixed wage recommendation. However, using a "rule-based" market model, we had to adopt the RB model because of the first edge. Thus, we cannot model this second edge appropriately as flexible. In our general DE market model, we are able to model the first edge as rigid and the second as flexible. In fact, we can define each edge independently of the players involved in the DE model.

Let us present another situation that no "rule-based" market model can account for. We consider two flexible workers and a firm p offering these two workers a job. The firm principally wants to contract on the salary. This is what happens in the contract of this firm with worker 1. Moreover, let worker 2 be a real good match for the firm p. Flexibly contracting with him however, would possibly disrupt the firm's wage-hierarchy structure. Thus, we would like to define the edges $(p, 1)$ and $(p, 2)$ as flexible and rigid, respectively. This is clearly not possible with any "rule-based" market model but no problem at all for our general DE market model.

We give a last example for the shortcoming of any "rule-based" market model and hereby an example illustrating the usefulness of our decisive edges model. Let all workers in this example be flexible. Note that not all labour unions are equally strong. Thus, there are minimum wages for some jobs while there are no minimum wages for others.[17] Imagine that firm 1 offers a job to worker 1 where a minimum wage applies while there is no minimum wage applying to the job offer of firm 2 to any worker. Clearly, we will have to define firm 1 as rigid and firm 2 as flexible in a "rule-based" market model, since we want the edges $(1, 1)$ and $(2, 1)$ to be rigid and flexible, respectively. Moreover, we have to define an edge as rigid if at least one of the two players involved is rigid (adopting the RB model).[18] Having all this set, we cannot model the situation where firm 1 (rigid player) offers a job to worker 2 where no minimum wage applies.[19]

[15] Highly riskaverse and slightly riskaverse workers would be another distinction that leads to the same conclusion.

[16] Note that this concerns both sides of the labour market, since a human resource chief of a certain firm can be a member of a labour union or not.

[17] Alternatively, we could imagine strong wage recommendations for some jobs and weak wage recommendations for other jobs.

[18] Note that all definitions have been necessary and unique.

[19] We can think of several reasons why there should not be a minimum wage applying to the edge $(1, 2)$. The

Again, the DE market model can describe this situation appropriately, while any "rule-based" market model like the RB or the FB market model cannot.

Besides all the mentioned economic reasons, there is of course genuine mathematical interest for our generalisation.

The following chapter is devoted to the main task of the thesis: It presents our modified auction algorithm and exploits it to prove that there exists a stable outcome in the general DE market model.

job offer of firm 1 to worker 2 can be different from the one to worker 1, since worker 1 and 2 differ in capabilities, education, physical condition, domicile, marital status, disablement and so on.

Chapter 4

Modified Auction Algorithm in the DE Model

Eriksson and Karlander (2000) provided a pseudo-polynomial auction algorithm to prove the existence of stable outcomes in their mixed matching market model, which we call the RB market model. Based on the ideas of Eriksson and Karlander (2000), Hochstättler et al. (2006) constructed a polynomial auction algorithm and proved that it runs in $\mathcal{O}\left(n^4\right)$ time where $2n$ denotes the number of players. In this chapter we first give some necessary definitions. Afterwards, we present a modification of the auction algorithm of Hochstättler et al. (2006). We then establish its correctness in our general DE market model[1] and hereby prove the existence of a stable outcome in the DE market model. Moreover, we show that our modified auction algorithm runs in $\mathcal{O}\left(n^4\right)$ time, too. Finally, we close the chapter (and thereby the thesis) by giving concrete examples that show how our algorithm works and why our modifications to the auction algorithm of Hochstättler et al. (2006) are necessary.

4.1 Some Definitions

As previously mentioned, we will assume that $|P| = |Q| = n$, since we can always introduce dummy nodes with zero edge weights. Let $\mu : P \to Q$ be a partial map and let i and j denote the index for firms and workers, respectively. If $\mu(i) = j$, then we say that firm i **proposes** to worker j. A proposal is called **flexible** or **rigid** if the corresponding edge is flexible or rigid. Moreover, we call a firm i (a worker j) **mapped** if $i \in \mu^{-1}(Q)$ (if $j \in \mu(P)$) and **unmapped** otherwise. If $\mu(i_1) = \mu(i_2) = j$ with $i_1 \neq i_2$, then j is called **doubly mapped**.[2] Furthermore, we recall the definition of a **payoff** as a pair (u, v) with the vectors $u, v \in \mathbb{R}^n$. Moreover, we use the following notation:

P_U	set of unmapped firms
$Q_{2\mu}$	set of doubly mapped workers
Q_R	set of rigidly mapped workers
Q_{2R}	set of workers with at least two rigid proposals

[1] See section 3.4.

[2] Note that any multiply mapped worker is contained in this definition of a doubly mapped worker, i.e. a triply or quadruply mapped worker etc.

Furthermore, we define

$$f_{ij}^{(v;\mu)} := \begin{cases} a_{ij} + b_{ij} - v_j & \text{if } (i,j) \text{ is a flexible edge} \\ a_{ij} & \text{if } (i,j) \text{ is rigid and } v_j < b_{ij} \\ a_{ij} & \text{if } (i,j) \text{ is rigid and } v_j = b_{ij} \text{ and } \mu(i) = j \\ 0 & \text{otherwise.} \end{cases} \tag{4.1}$$

Obviously, $f_{ij}^{(v;\mu)}$ denotes the possible profit firm i receives from a contract with the worker j given the workers' payoff vector v. Note that $f_{ij}^{(v;\mu)}$ additionally depends on the mapping μ.[3] We define the **augmentation digraph** $G^{(v;\mu)}$ as the directed subgraph of G with the arc set

$$A^{(v;\mu)} := \{(j,i) \mid j = \mu(i)\} \cup \left\{(i,j) \mid j \in D_i^{(v;\mu)}\right\}, \tag{4.2}$$

where

$$D_i^{(v;\mu)} := \left\{ j \in Q \mid f_{ij}^{(v;\mu)} = \max_k f_{ik}^{(v;\mu)} \right\}. \tag{4.3}$$

Note that $D_i^{(v;\mu)}$ is the set of workers that maximise the potential benefit of firm i. Thus, $D_i^{(v;\mu)}$ contains the favourite partners of the firm i given the current workers' payoff v and the current mapping μ. In the augmentation digraph $G^{(v;\mu)}$ we therefore have mapping backward arcs and forward arcs that end in a favourite partner of the corresponding firm. A directed path $\mathcal{P}$ in $G^{(v;\mu)}$ that connects a doubly mapped worker $j_1 \in Q_{2\mu}$ with another worker j_l is called (μ-) **alternating**. If j_l is unmapped, then the path $\mathcal{P}$ is called (μ-) **augmenting**. Lastly, we let $BFS\left(G^{(v;\mu)}, Q_{2\mu}\right)$ denote a procedure that implements a breadth first search[4] in the augmentation digraph $G^{(v;\mu)}$. It returns all nodes that are reachable in $G^{(v;\mu)}$ from the set $Q_{2\mu}$.

We are now ready to give the definition of our modified auction algorithm in the next section.

4.2 The Modified Auction Algorithm

We first present the main part of our modified auction algorithm.[5]

Algorithm 4.1: Modified Procedure *Construction of a Stable Outcome*

$v \leftarrow 0$
PLACERIGIDPROPOSALS
while $Q_{2\mu} \neq \emptyset$ **do**
    while there is a μ-alternating path $\mathcal{P}$ to $j \in (Q \backslash \mu(P)) \cup Q_R$ **do**
        DISPOSERIGID(j)
        ALTERNATE($\mathcal{P}$)
        PLACERIGIDPROPOSALS
    end while
    HUNGARIANUPDATE
end while
$u_i \leftarrow \max_j f_{ij}^{(v;\mu)}$

[3] See line 3 of the definition of $f_{ij}^{(v;\mu)}$ in (4.1).

[4] See section 2.4.

[5] Recall that we give a modification of the auction algorithm of Hochstättler et al. (2006). Thus, the terms modified and unmodified must be understood relative to their algorithm.

We next give the exploited algorithms PLACERIGIDPROPOSALS and HUNGARIANUPDATE.

Algorithm 4.2: Modified Procedure PLACERIGIDPROPOSALS

do
    for all $i \in P_U$ **do**
        PROPOSE(i)
    end for
    for all $j \in Q_R$ **do**
        Let i^* be the favourite rigid proposal in $\mu^{-1}(j)$
        if $v_j < b_{i^*j}$ **then**
            $\mu^{-1}(j) := \{i^*\}$
            $v_j \leftarrow b_{i^*j}$
    end for
while $P_U \neq \emptyset$

Algorithm 4.3: Unmodified Procedure HUNGARIANUPDATE

$\bar{P}\dot{\cup}\bar{Q} \leftarrow \mathrm{BFS}\left(G^{(v;\mu)}, Q_{2\mu}\right)$
$u_i \leftarrow \max_j f_{ij}^{(v;\mu)}$
$\Delta \leftarrow \min\left\{u_i - f_{iz}^{(v;\mu)} \mid i \in \bar{P},\, z \in Q \backslash \bar{Q}\right\}$
for all $j \in \bar{Q}$ **do**
    $v_j \leftarrow v_j + \Delta$
end for

Lastly, we briefly explain the exploited subroutines.

ALTERNATE($\mathcal{P}$) gets an alternating path as argument and reverses the orientation of all arcs along this path. The map μ is modified s.t. it contains the new backward arcs.

BFS$\left(G^{(v;\mu)}, Q_{2\mu}\right)$ determines all nodes that are reachable from a doubly mapped worker in the augmentation digraph $G^{(v;\mu)}$ exploiting a breadth first search.

DISPOSERIGID(j) disposes a possibly existing rigid proposal of the worker j. Thus, we undefine $\mu(i)$, if we have a firm $i \in \mu^{-1}(j)$ with $(i,j) \in R^*$.

PROPOSE(i) places a proposal from firm i to a worker in $D_i^{(v;\mu)}$, i.e. it chooses $\mu(i) \in D_i^{(v;\mu)}$. If there is a rigid proposal possible, then it is favoured over flexible proposals, i.e. if there exists a worker $j \in D_i^{(v;\mu)}$ with $(i,j) \in R^*$, then i rigidly proposes to such a j.

4.3 Correctness of the Modified Auction Algorithm

Before proving the correctness of the modified auction algorithm 4.1, we want to develop some intuition by commenting on what the algorithm mainly does. The map μ always defines stable relations but it is not necessarily injective. During the algorithm we will increase $|\mu(P)|$ until the map becomes injective. The procedure to increase $|\mu(P)|$ works on the augmentation digraph $G^{(v;\mu)}$. In the beginning of the algorithm 4.1 we set the payoff of all workers at zero. Moreover, we choose a map μ s.t. $\mu(i) \in D_i^{(v;\mu)}$ for all $i \in P$.[6] For all rigidly mapped workers we

[6]Note that this is possible, since the set $D_i^{(v;\mu)}$ is nonempty for all $i \in P$.

choose the best rigid proposal, dispose all other proposals in the first place and set the workers' payoff accordingly. Thus, there can be firms that are temporarily unmapped. Such firms keep proposing until every firm is mapped. However, disposed rigid proposals will never be proposed again. In contrast to this, disposed flexible proposals might be proposed again in case they are still optimal for the proposing firm after the increase in the workers' payoff. Hence, this part of the algorithm is analoguous to the famous "men-propose-women-dispose" algorithm of Gale and Shapley (1962) what concerns rigid edges[7]: Every worker with more than one rigid proposal chooses the best one and disposes all others. Taken together, we ensure that there is no unmapped firm, that every worker has at most one rigid proposal (the best rigid proposal), that the workers' payoff is updated accordingly and that still optimal flexible proposals are kept. The algorithm then searches for alternating paths in the augmentation digraph that lead to an unmapped or a rigidly mapped worker. If there is an alternating path to an unmapped worker, then we are able to subsequently increase the size of the mapping image. If there is an alternating path to a rigidly mapped worker, then we can dispose a rigid proposal (that will never be proposed again) without decreasing the size of the mapping image. Finally, if there is no alternating path to an unmapped or a rigidly mapped worker, then the payoff of the workers which are reachable in $G^{(v;\mu)}$ from the set $Q_{2\mu}$ are increased by Δ. This leads to at least one new arc in the augmentation digraph. We repeat this procedure until we obtain a path in the augmentation digraph $G^{(v;\mu)}$ as desired. The algorithm finally terminates as soon as there is no doubly mapped worker anymore, which means that the map μ has become injective.

The following lemma will be exploited in the correctness proof.

LEMMA 4.4

We consider the modified auction algorithm 4.1 in the DE market model. Then the following statements hold.

(i) The procedure PLACERIGIDPROPOSALS *never decreases* $|\mu(P)|$.

(ii) The payoff vector v *increases monotonically.*

(iii) A disposed rigid proposal will never be proposed again.

Proof:

(i) Let us prove the first statement. There is only one place where the procedure PLACE-RIGIDPROPOSALS could decrease $|\mu(P)|$: Line 8. In line 8 the size of the mapping image remains the same because we dispose all proposals except for the best rigid proposal to a worker in Q_R.

(ii) We next show that the second statement holds. Initially, the workers' benefit is set at zero. There are only two places in the entire algorithm where the payoff vector v changes. First, we discuss line 9 of the procedure PLACERIGIDPROPOSALS. Here, we set the payoff of newly rigidly mapped workers. A rigid proposal (i, j) however, can only be made if $v_j < b_{ij}$ because j must be a favourite partner of i, i.e. $j \in D_i^{(v;\mu)}$. Thus, setting $v_j = b_{ij}$

[7] We will see this more clearly later in example 4.7 where we solve an instance of the marriage problem.

strictly increases the worker's benefit. The second and last place is line 5 in the procedure HUNGARIANUPDATE where we strictly increase v, as Δ can be shown to be strictly positive.

(iii) Finally, we prove the last statement. We note that the procedure PLACERIGIDPROPOSALS ensures that we have $v_j = b_{ij}$ for all rigid proposals (i,j). Obviously, an unmapped rigid edge (i,j) can only become a rigid proposal if it is in the augmentation digraph, which means that we must have $v_j < b_{ij}$. This however, cannot happen if the rigid edge (i,j) has already been proposed, since v_j increases monotonically according to part (ii). ■

We are now ready to discuss the correctness of the modified auction algorithm 4.1.

THEOREM 4.5 (CORRECTNESS OF THE MODIFIED AUCTION ALGORITHM 4.1)
The modified auction algorithm 4.1 produces a stable outcome in the DE market model.

Proof:

a) In all iterations of the inner while loop the modified auction algorithm 4.1 either increases $|\mu(P)|$ or disposes a rigid proposal. This can be seen from the following two cases.

   1. If there is an alternating path $\mathcal{P}$ to an unmapped worker j (augmenting path), then DISPOSERIGID(j) trivially does not change anything. Afterwards, the subroutine ALTERNATE($\mathcal{P}$) increases $|\mu(P)|$. Moreover, ALTERNATE($\mathcal{P}$) possibly transforms some rigid forward arcs into rigid proposals. Finally, the procedure PLACERIGIDPROPOSALS gets called. This procedure might set some rigidly mapped workers' benefit and therefore perform some other steps, too. However, we know from part (i) of lemma 4.4 that it never decreases $|\mu(P)|$, which means that size of mapping image has increased after the run of the inner while loop.
   2. If there is an alternating path $\mathcal{P}$ to a worker j in Q_R, then DISPOSERIGID(j) removes the current rigid proposal (i,j) of j where $i \in \mu^{-1}(j)$ and $(i,j) \in R^*$. This disposed rigid edge will never be proposed again according to part (iii) of lemma 4.4. Afterwards, the procedures ALTERNATE($\mathcal{P}$) and PLACERIGIDPROPOSALS get called. ALTERNATE($\mathcal{P}$) leaves $|\mu(P)|$ unchanged for obvious reasons. Lastly, we know from part (i) of lemma 4.4 that PLACERIGIDPROPOSALS never decreases $|\mu(P)|$. Taken together, we dispose a rigid proposal without changing $|\mu(P)|$ in this case.

b) If there is no alternating path $\mathcal{P}$ to a worker $j \in (Q \backslash \mu(P)) \cup Q_R$, then the procedure HUNGARIANUPDATE is called. We will first show that each call of HUNGARIANUPDATE leads to a new arc in the augmentation digraph. In the procedure HUNGARIANUPDATE we begin by determining the set of firms $\bar{P}$ and of workers $\bar{Q}$, which are reachable from the set of doubly mapped workers in the augmentation digraph. Thus, there is an alternating path to every firm in $\bar{P}$ and every worker in $\bar{Q}$. We therefore have $((Q \backslash \mu(P)) \cup Q_R) \cap \bar{Q} = \emptyset$, since the condition of the inner while loop is not satisfied when the procedure HUNGARIANUPDATE gets called. We then set the payoff of the firms at the best currently possible value and compute the quantity Δ. Recall that $\Delta = \min\left\{u_i - f_{iz}^{(v;\mu)} \mid i \in \bar{P} \text{ and } z \in Q \backslash \bar{Q}\right\}$. Let

$\Delta = u_{i^*} - f_{i^* z^*}^{(v;\mu)}$ for some $i^* \in \bar{P}$ and $z^* \in Q \backslash \bar{Q}$. The last step of the procedure HUNGARIANUPDATE is to increase the benefit of all workers in $\bar{Q}$ by Δ. We give the following useful statements.

(i) If $i \in \bar{P}$, then $D_i^{(v;\mu)} \subset \bar{Q}$.

(ii) If $j = \mu(i)$, then $j \in \bar{Q}$ if and only if $i \in \bar{P}$.

(iii) If $j = \mu(i)$ and $i \in \bar{P}$, then $j \notin Q_R$ because $j \in \bar{Q}$ according to (ii).

(iv) We have $(i, j) \in F^*$ for all $i \in \bar{P}$ and $j \in D_i^{(v;\mu)}$ because $(i, \mu(i)) \in F^*$ according to (iii) and because of the definition of the subroutine PROPOSE(i).

It follows from (iv) that $(i^*, j) \in F^*$ for all $j \in D_{i^*}^{(v;\mu)}$. Note that we increase the payoff of all workers in $\bar{Q}$ by Δ, which means that we especially increase the payoff of every $j \in D_{i^*}^{(v;\mu)}$ according to (i). Thus, we obtain $z^* \in D_{i^*}^{(v;\mu)}$ and therefore the new forward arc (i^*, z^*) in the augmentation digraph as claimed in the beginning of the current part b). This new forward arc might lead to an alternating path to a worker $j \in (Q \backslash \mu(P)) \cup Q_R$ in the resulting augmentation digraph or not. Clearly, the procedure HUNGARIANUPDATE gets called as long as we do not get a desired path and with each call we obtain a new arc. This process of always obtaining a new arc in the augmentation digraph[8] eventually provides an alternating path to $j \in Q \backslash \mu(P) \cup Q_R$. Hence, the inner while loop condition will finally be satisfied.[9]

c) Note that we know from part a) that each run of the inner while loop of the main algorithm 4.1 either increases $|\mu(P)|$ or disposes a rigid proposal. Obviously, $|\mu(P)|$ can increase at most n times while we can dispose at most n^2 rigid proposals, since we know from part (iii) of lemma 4.4 that a disposed rigid proposal will never be proposed again. Moreover, we have seen in part b) that it takes at most n^2 calls of the procedure HUNGARIANUPDATE until the next run of the inner while loop. Hence, the algorithm is finite.

d) It remains to show that the modified auction algorithm 4.1 produces a stable outcome. Let us define $\bar{u}_i := \max_j f_{ij}^{(v;\mu)}$. The payoff $(\bar{u}, v)$ then is stable at any stage of the algorithm, since no firm will ever form a blocking pair. Moreover, we note that the subroutines PROPOSE(i) and ALTERNATE($\mathcal{P}$) imply that $j \in D_i^{(v;\mu)}$ if $j = \mu(i)$. The fact that μ must be bijective at the end of the algorithm[10] and the definition of $f_{ij}^{(v;\mu)}$ in (4.1) then ensure that $(u, v; \mu)$ satisfies conditions (iv) and (v) of the stability definition 3.5 because we set $u_i := \max_j f_{ij}^{(v;\mu)}$ in the very end of the algorithm. We next prove the individual rationality of the outcome. We have $v \geq 0$, as we initially set v at zero and since it increases monotonically according to part (ii) of lemma 4.4. We next prove that we also have $u \geq 0$. Note that the outer while loop ensures that there is at least one doubly mapped worker, when the main part of the algorithm runs. Because $|P| = |Q| = n$ and the fact that the mapping size never decreases (a once mapped worker never gets unmapped again)[11], we

[8] Obviously, we can obtain at most n^2 new arcs.

[9] Note that the condition of the outer while loop ensures that there is at least one doubly mapped worker. The fact that each firm is mapped to at most one worker and $|P| = |Q| = n$ let us therefore conclude that there must be at least one unmapped worker.

[10] We will show this in the very end of the proof.

[11] See part a) of the proof.

can conclude that we always have an unmapped worker until the very end of the algorithm. An unmapped worker j however, is of nonnegative value for firm i because we have $v_j = 0$ and therefore $f_{ij}^{(v;\mu)} = a_{ij} + b_{ij} - v_j \geq 0$ if (i, j) is flexible and $f_{ij}^{(v;\mu)} = a_{ij} \geq 0$ if (i, j) is rigid with the nonnegativity of the matrices a and b. This clearly holds for all firms. Finally, we note that we set $u_i := \max_j f_{ij}^{(v;\mu)}$ in the very end of the algorithm. Thus, we have $u \geq 0$ as claimed. We know from part c) that the algorithm is finite. The condition of the outer while loop of the main algorithm 4.1 ensures that there are no doubly mapped workers in the end. This and the fact that we have $|P| = |Q| = n$ imply that the map μ has become bijective. Hence, $(u, v; \mu)$ is a stable outcome with definition 3.5, which completes the proof. ■

4.4 Complexity of the Modified Auction Algorithm

We give the following theorem.

Theorem 4.6 (Complexity of the Modified Auction Algorithm 4.1)
The modified auction algorithm 4.1 runs in $\mathcal{O}\left(n^4\right)$ time in the DE market model.

Proof:
Generally, we exploit the complexity proof of Hochstättler et al. (2006, Theorem 1, p. 4-6) and some insights from the proof of theorem 4.5.

Note that the inner while loop of the main algorithm 4.1 is executed at most n^2+n times, since $|\mu\left(P\right)|$ increases at most n times and because there at most n^2 rigid proposals to dispose.[12] Thus, the number of executions of the inner while loop is in $\mathcal{O}\left(n^2\right)$. Moreover, each call of the procedure HungarianUpdate adds at least one new arc to the augmentation digraph. This means that we must have an alternating path to a worker $j \in (Q \backslash \mu\left(P\right)) \cup Q_R$ after at most n^2 calls of this procedure. We use a standard implementation of the procedure HungarianUpdate[13], which ensures that the consecutive calls of the procedure until a desired path is found need $\mathcal{O}\left(n^2\right)$ time in total. This includes an update of the augmentation digraph by reusing the BFS-structure from the previous call and storing a minimum distance $\Delta_j = \min\left\{u_i - f_{ij}^{(v;\mu)} \mid i \in \bar{P}\right\}$ from a worker $j \in Q\backslash\bar{Q}$ to $\bar{P} \cup \bar{Q}$. The quantity Δ_j can be updated for each worker j in $\mathcal{O}\left(n\right)$ time. Moreover, we have to update them each time we add a node to $\bar{P} \cup \bar{Q}$, which trivially happens $\mathcal{O}\left(n\right)$ times. We can then compute $\Delta = \min_j \{\Delta_j\}$ in $\mathcal{O}\left(n\right)$ time. After an update of the payoffs in line 5 of the procedure HungarianUpdate, we set $\Delta_j \leftarrow \Delta_j - \Delta$. This way, HungarianUpdate can be implemented to continue the BFS of its previous call with modified payoffs. Thus, the accumulated time spent is $\mathcal{O}\left(n^2\right)$. Afterwards, we have found an alternating path to an unmapped or rigidly mapped worker, which means that we can then increase the mapping size image or dispose a rigid proposal. Taken together, we get a complexity of our modified auction algorithm 4.1 of at least $\mathcal{O}\left(n^4\right)$. We will next prove that it will not be greater.

To this end, it clearly remains to show that the total complexity of the inner while body is no greater than $\mathcal{O}\left(n^4\right)$. The total complexity of the subroutines DisposeRigid(j) and Alternate$(\mathcal{P})$ is obviously bounded by $\mathcal{O}\left(n^2\right)$ and $\mathcal{O}\left(n^3\right)$, respectively. Lastly, the procedure

[12] Recall that a disposed rigid proposal will never be proposed again.
[13] The interested reader is referred to Galil (1986) and Hochstättler et al. (2005) for more details.

that deserves more attention is PLACERIGIDPROPOSALS. In the first call, we must make n proposals. Note that we have to find a favourite partner for each firm $i \in P$. We have n firms and n workers and hence, the complexity is $\mathcal{O}\left(n^2\right)$ so far. However, the sets $D_i^{(v;\mu)}$ can change during the algorithm, which means that we have to update the augmentation digraph. Note that the procedure PLACERIGIDPROPOSALS in the inner while loop of the main algorithm 4.1 gets called $\mathcal{O}\left(n^2\right)$ times. Thus, we can allocate the needed $\mathcal{O}\left(n^2\right)$ time to update the augmentation digraph without changing the claimed complexity of the entire modifed auction algorithm 4.1, which completes the proof. ■

With the complexity of the modified auction algorithm 4.1 in the DE market model, we clearly get an upper bound for the complexity in any "rule-based" market model, since these are just special cases. We thus conclude that the modified auction algorithm 4.1 runs in in $\mathcal{O}\left(n^4\right)$ time in, for instance, the RB and the FB market model. This is a new result for the FB market model but not for the RB market model, as the auction algorithm of Hochstättler et al. (2006) also displays a complexity of $\mathcal{O}\left(n^4\right)$ in the RB market model of Eriksson and Karlander (2000).

4.5 Comparison to the Algorithm of Hochstättler et al. (2006)

As previously mentioned, our algorithm 4.1 represents a modification of the auction algorithm of Hochstättler et al. (2006). Hochstättler et al. (2006) proved that their algorithm finds a stable outcome in the RB model. We have corrected some minor mistakes and we have modified their algorithm s.t. it finds a stable outcome in our more general DE model. We now shortly comment on these modifications.

The first modification concerns the inner while loop of the main algorithm 4.1. We additionally introduce the subroutine DISPOSERIGID(j) to dispose the rigid proposal in case we have an alternating path to a worker in Q_R. Furthermore, we update the firms' benefit in the very last line 11 of the main algorithm 4.1, since we want to account for the possibility that the outer while loop is never executed.[14]

Moreover we have changed the subroutine PROPOSE(i) to favour rigid proposals over flexible ones. This ensures that the correctness proof works. In contrast, we did not have to modify the procedure HUNGARIANUPDATE at all.

Lastly, we have modified the procedure PLACERIGIDPROPOSALS. First, we define it as a do-while-loop instead of a while-do-loop to ensure that its body is executed at least once. We make this adjustment, as there can be workers in Q_R or even in Q_{2R} when there are no unmapped firms at all. Finally, we select the best rigid proposal for all workers $j \in Q_R$. If the payoff of such a worker is not updated yet, then we dispose all other proposals and consequently adjust the payoff. All disposed rigid proposals will never be proposed again as we have proved in part (iii) of lemma 4.4. Disposed flexible proposals however, might be proposed again when we still have $j \in D_i^{(v;\mu)}$ (for $(i,j) \in F^*$) after the payoff increase of v_j. This will happen in the next while loop. But this time we will not dispose the reproposed flexible edges, since we now do not enter the if-branch. Hence, taken together, the entire modified procedure PLACERIGIDPROPOSALS ensures that there is no unmapped firm and that each worker has at most one rigid proposal, it

[14] The marriage problem is an example where the outer while loop is never run. Another example is that each firm proposes to a different worker in the first call of the procedure PLACERIGIDPROPOSALS.

sets the payoffs of all newly rigidly mapped workers and unmaps flexible proposals which have become unattractive to the proposing firm because the benefit vector v has increased.[15]

4.6 Some Examples

We close the thesis by presenting three examples. In the first example, we want to show how our modified auction algorithm 4.1 works in the special case of the marriage problem. The second example addresses the same issue in the special case of the assignment game. Finally, the third example is devoted to a pure DE market setting that cannot be dealt with in a RB or FB market model. We demonstrate that the auction algorithm of Hochstättler et al. (2006) breaks down in this last example while our modified auction algorithm 4.1 produces a stable outcome as claimed in theorem 4.5.

In all the following examples, we display all players with striped circles. Moreover, we mark the weight entries of rigid and flexible edges in the matrices a and b in normal and bold face, respectively. Furthermore, we always highlight rigid edges with the letter "R" in the augmentation digraph and in the mapping μ. Finally, we will not draw the forward arcs to mapped workers for expositional ease. Note that there would always be a forward arc (i, j) to every backward (mapping) arc (j, i) because we have $j \in D_i^{(v;\mu)}$ if $j = \mu(i)$.

Example 4.7
We consider

$$\begin{aligned} P &= \{p_1, p_2, p_3, p_4, p_5\} \\ Q &= \{q_1, q_2, q_3, q_4, q_5\} \\ R^* &= P \times Q \text{ and } F^* = \emptyset \end{aligned}$$

We are clearly reduced to the marriage problem, since we only have rigid edges. The weight matrices a and b are given below.

a_{ij}	q_1	q_2	q_3	q_4	q_5
p_1	3	2	3	2	4
p_2	1	3	2	3	2
p_3	4	2	4	1	4
p_4	2	3	2	3	2
p_5	4	4	2	1	4

Table 4.1: Weight matrix a in example 4.7.

[15] We note that we have modified everything except for the lines 2 to 4 (where the unmapped firms make their proposals) of the procedure PLACERIGIDPROPOSALS compared to the auction algorithm of Hochstättler et al. (2006).

b_{ij}	q_1	q_2	q_3	q_4	q_5
p_1	3	2	3	3	2
p_2	2	3	1	2	2
p_3	2	1	3	2	3
p_4	3	3	3	2	4
p_5	2	3	2	1	3

Table 4.2: Weight matrix b in the example 4.7.

1. First step:

   We set the payoff of all workers at zero and call the procedure PLACERIGIDPROPOSALS. We first compute the potential payoffs of the firms $f_{ij}^{(v;\mu)}$ with (4.1) and then compute the sets $D_i^{(v;\mu)}$ using (4.3). We get

$f_{ij}^{(v;\mu)}$	q_1	q_2	q_3	q_4	q_5
p_1	3	2	3	2	4
p_2	1	3	2	3	2
p_3	4	2	4	1	4
p_4	2	3	2	3	2
p_5	4	4	2	1	4

   and

$$\begin{aligned} D_{p_1}^{(v;\mu)} &= \{q_5\} \\ D_{p_2}^{(v;\mu)} &= \{q_2, q_4\} \\ D_{p_3}^{(v;\mu)} &= \{q_1, q_3, q_5\} \\ D_{p_4}^{(v;\mu)} &= \{q_2, q_4\} \\ D_{p_5}^{(v;\mu)} &= \{q_1, q_2, q_5\}\,. \end{aligned}$$

   We generally adopt the rule of selecting the lowest index node in case we have to choose among several.[16] Hence, we have the map $\mu = \{1 \to 5, 2 \to 2, 3 \to 1, 4 \to 2, 5 \to 1\}$. We then get $Q_R = \{1, 2, 5\}$ and $Q_{2R} = \{1, 2\}$. Note that worker 1 has a rigid proposal from firm 3 and 5 and that we have $b_{31} = b_{51} = 2$. Again, we select the lowest index node. Thus, worker 1 chooses the best rigid[17] proposal $(3, 1)$, disposes firm 5 and we set $v_1 = b_{31} = 2$ (if-branch). Worker 2 also has two proposals. He chooses firm 2, disposes firm 4 and hence, we set $v_2 = b_{22} = 3$ (if-branch). Finally, worker 5 only has one proposal of firm 1, which is why we set $v_5 = b_{15} = 2$ (if-branch). In the next iteration, the two currently unmapped firms 4 and 5 propose to workers 4 and 5, respectively. Worker 4 only has one propsal and we therefore set $v_4 = b_{44} = 2$ (if-branch). Worker 5 however, has two proposals, i.e. an old proposal from firm 1 and a new one from firm 5. It chooses the best proposal, which stems from firm 5. Thus, worker 5 chooses firm 5, disposes firm 1 and we set $v_5 = b_{55} = 3$ (if-branch).[18] In the next iteration, the only unmapped firm 1 proposes to worker 1. This worker keeps firm 1, disposes firm 3 and we set $v_1 = b_{11} = 3$ (if-branch). In the following run of the while-body, the only unmapped firm 3 proposes to worker 3 who has been unmapped before. We therefore set $v_3 = b_{33} = 3$. The procedure PLACERIGIDPROPOSALS

[16] Strictly speaking, we should have defined our modified auction algorithm 4.1 with the mentioned rule to ensure uniqueness, since this is a necessary characteristic of any algorithm. However, we chose not to do so for the sake of brevity and the clarity of the exposition.

[17] Of course, we only have rigid proposals in the current instance of the marriage problem. We will therefore no longer mention the rigidity of edges in the remainder of the current example.

[18] Note that we again have $1, 2 \in Q_R$ in this iteration. However, there is nothing to do here because all these workers only have one old proposal. Their payoff is already set accordingly and there is no proposal to dispose. Note that the if-branch is not executed this time and hence, the algorithm does not change anything as desired. We will skip similar comments in the remainder to ease the exposition.

now terminates, as we do not have an unmapped firm at the moment. Summarising, we have

$\mu = \{1 \to 1, 2 \to 2, 3 \to 3, 4 \to 4, 5 \to 5\}$ $\quad D_{p_1}^{(v;\mu)} = \{q_1\}$
$\bar{u} = (3, 3, 4, 3, 4)$ $\quad D_{p_2}^{(v;\mu)} = \{q_2\}$
$u = (0, 0, 0, 0, 0)$ $\quad D_{p_3}^{(v;\mu)} = \{q_3\}$
$v = (3, 3, 3, 2, 3)$ $\quad D_{p_4}^{(v;\mu)} = \{q_4\}$
$Q_R = \{1, 2, 3, 4, 5\}$, $Q_{2\mu} = Q \backslash \mu(P) = \emptyset$ $\quad D_{p_5}^{(v;\mu)} = \{q_5\}$

and figure 4.1.

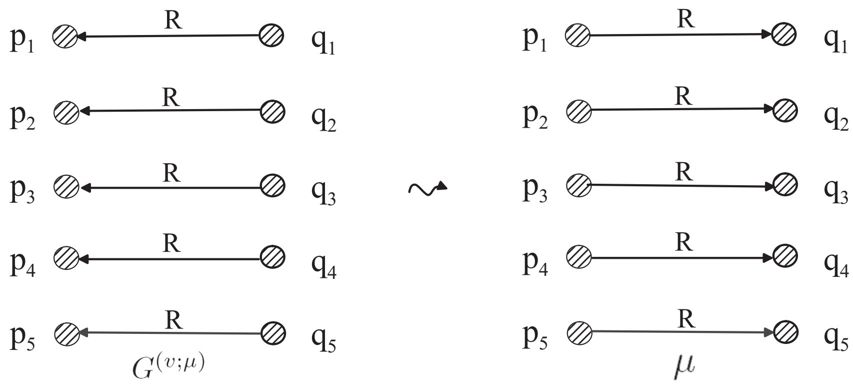

Figure 4.1: First step in example 4.7.

2. Second step:

   The condition of the outer while loop is not satisfied, since there is no doubly mapped worker. Thus, we finally have to execute line 11 of the main algorithm 4.1. We then set $u_1 = 3$, $u_2 = 3$, $u_3 = 4$, $u_4 = 3$ and $u_5 = 4$. This terminates the algorithm. Hence, we have reached a stable outcome $(u, v; \mu)$ with

   $\mu = \{1 \to 1, 2 \to 2, 3 \to 3, 4 \to 4, 5 \to 5\}$
   $\bar{u} = (3, 3, 4, 3, 4)$
   $u = (3, 3, 4, 3, 4)$
   $v = (3, 3, 3, 2, 3)$

The reader should convince himself that the above is really a stable outcome.[19] We note that the procedure PLACERIGIDPROPOSALS essentially reduces to the famous "men-propose-women-dispose" algorithm of Gale and Shapley (1962) in the special case of the marriage problem.

[19] See definition 3.5 of a stable payoff and outcome in the DE model or alternatively, the stability definition 3.4 in the marriage problem.

Example 4.8
We consider

$$\begin{aligned} P &= \{p_1, p_2, p_3, p_4, p_5\} \\ Q &= \{q_1, q_2, q_3, q_4, q_5\} \\ R^* &= \emptyset \text{ and } F^* = P \times Q \end{aligned}$$

We are obviously reduced to the assignment game, since there are only flexible edges. The weight matrix $a = b$ is given below.

a_{ij}	q_1	q_2	q_3	q_4	q_5
p_1	**3**	**2**	**3**	**2.5**	**3**
p_2	**1.5**	**3**	**1.5**	**2.5**	**2**
p_3	**3**	**1.5**	**3**	**1.5**	**3.5**
p_4	**2.5**	**3**	**2.5**	**2.5**	**3**
p_5	**3**	**3.5**	**2**	**1**	**2**

Table 4.3: Weight matrix $a = b$ in example 4.8.

1. First step:

   We set the payoff of all workers at zero and call the procedure PLACERIGIDPROPOSALS. We first compute the potential payoffs of the firms $f_{ij}^{(v;\mu)}$ with (4.1) and then compute the sets $D_i^{(v;\mu)}$ using (4.3). We get

$f_{ij}^{(v;\mu)}$	q_1	q_2	q_3	q_4	q_5
p_1	6	4	6	5	6
p_2	3	6	3	5	4
p_3	6	3	6	3	7
p_4	5	6	5	5	6
p_5	6	7	4	2	4

   and

   $$\begin{aligned} D_{p_1}^{(v;\mu)} &= \{q_1, q_3, q_5\} \\ D_{p_2}^{(v;\mu)} &= \{q_2\} \\ D_{p_3}^{(v;\mu)} &= \{q_5\} \\ D_{p_4}^{(v;\mu)} &= \{q_2, q_5\} \\ D_{p_5}^{(v;\mu)} &= \{q_2\} \end{aligned}$$

   Recall that we select the lowest index node in case we have to choose among several. Hence, we have the map $\mu = \{1 \to 1, 2 \to 2, 3 \to 5, 4 \to 2, 5 \to 2\}$. Trivially, we get $Q_R = Q_{2R} = \emptyset$, as we only have flexible edges in the current assignment game example. Thus, the procedure PLACERIGIDPROPOSALS already terminates, as we do not have an unmapped firm at the moment. Summarising, we have

   $\mu = \{1 \to 1, 2 \to 2, 3 \to 5, 4 \to 2, 5 \to 2\}$ $\quad$ $D_{p_1}^{(v;\mu)} = \{q_1, q_3, q_5\}$
   $\bar{u} = (6, 6, 7, 6, 7)$ $\quad$ $D_{p_2}^{(v;\mu)} = \{q_2\}$
   $u = (0, 0, 0, 0, 0)$ $\quad$ $D_{p_3}^{(v;\mu)} = \{q_5\}$
   $v = (0, 0, 0, 0, 0)$ $\quad$ $D_{p_4}^{(v;\mu)} = \{q_2, q_5\}$
   $Q_R = \emptyset$, $Q_{2\mu} = \{2\}$ and $Q \backslash \mu(P) = \{3, 4\}$ $\quad$ $D_{p_5}^{(v;\mu)} = \{q_2\}$

   and figure 4.2.

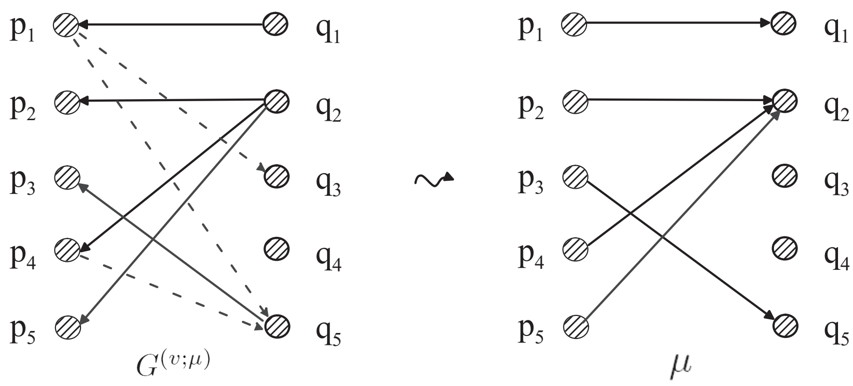

Figure 4.2: First step in example 4.8.

2. Second step:

   The condition of the outer while loop is satisfied with the doubly mapped worker 2. We do not have an alternating path $\mathcal{P}$ to a worker $j \in (Q \backslash \mu(P)) \cup Q_R$, i.e. there is no alternating path to an unmapped worker (no augmenting path) and we always have $Q_R = \emptyset$ in the current assignment game example. Thus, the procedure HungarianUpdate gets called. We obtain $\bar{P} = \{2,4,5\}$, $\bar{Q} = \{2,5\}$ and set $u_1 = 6$, $u_2 = 6$, $u_3 = 7$, $u_4 = 6$ and $u_5 = 7$. Moreover, we get $\Delta = 1$. Hence, we have to set $v_2 = 0 + 1 = 1$ and $v_5 = 0 + 1 = 1$. Note that this payoff update yields the new arcs $(2,4)$, $(3,1)$, $(3,3)$, $(4,1)$, $(4,3)$, $(4,4)$ and $(5,1)$ in the augmentation digraph. This completes the procedure HungarianUpdate.

   Summarising, we have

   $\mu = \{1 \to 1, 2 \to 2, 3 \to 5, 4 \to 2, 5 \to 2\}$
   $\bar{u} = (6,5,6,5,6)$
   $u = (6,6,7,6,7)$
   $v = (0,1,0,0,1)$
   $Q_R = \emptyset$, $Q_{2\mu} = \{2\}$ and $Q \backslash \mu(P) = \{3,4\}$

   $D_{p_1}^{(v;\mu)} = \{q_1, q_3\}$
   $D_{p_2}^{(v;\mu)} = \{q_2, q_4\}$
   $D_{p_3}^{(v;\mu)} = \{q_1, q_3, q_5\}$
   $D_{p_4}^{(v;\mu)} = \{q_1, q_2, q_3, q_4, q_5\}$
   $D_{p_5}^{(v;\mu)} = \{q_1, q_2\}$

   and figure 4.3.

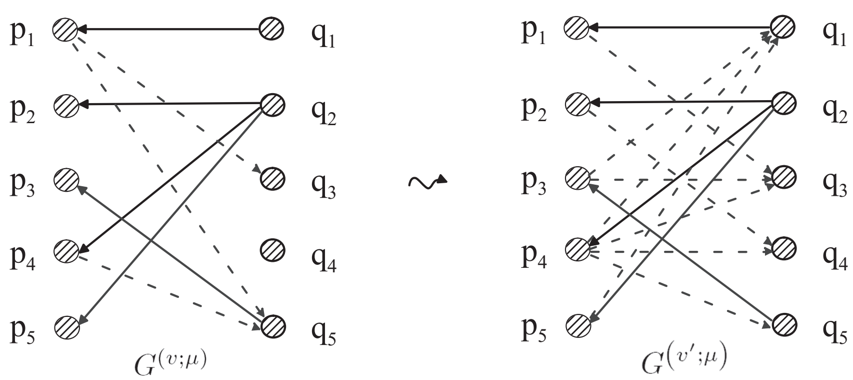

Figure 4.3: Second step in example 4.8.

3. Third step:

   The condition of the outer while loop is still satisfied with the doubly mapped worker 2. We now have several alternating paths to an unmapped worker.[20] Let us consider the alternating path $\mathcal{P} = (q_2, p_4, q_3)$ to the unmapped worker 3. In the inner while loop body, we first have to execute the subroutine DisposeRigid(3). This does not change anything, as the worker 3 is unmapped. Afterwards, we alternate the path $\mathcal{P}$. Note that this step increases the size of the mapping image as desired. Finally, the procedure PlaceRigidProposals gets called. However, nothing happens here, since there is trivially no rigidly mapped worker and because there is no unmapped firm at the moment. Summarising, we now have

   $\mu = \{1 \to 1, 2 \to 2, 3 \to 5, 4 \to 3, 5 \to 2\}$ $\quad D_{p_1}^{(v;\mu)} = \{q_1, q_3\}$
   $\bar{u} = (6, 5, 6, 5, 6)$ $\quad D_{p_2}^{(v;\mu)} = \{q_2, q_4\}$
   $u = (6, 6, 7, 6, 7)$ $\quad D_{p_3}^{(v;\mu)} = \{q_1, q_3, q_5\}$
   $v = (0, 1, 0, 0, 1)$ $\quad D_{p_4}^{(v;\mu)} = \{q_1, q_2, q_3, q_4, q_5\}$
   $Q_R = \emptyset$, $Q_{2\mu} = \{2\}$ and $Q \backslash \mu(P) = \{4\}$ $\quad D_{p_5}^{(v;\mu)} = \{q_1, q_2\}$

   and figure 4.4.

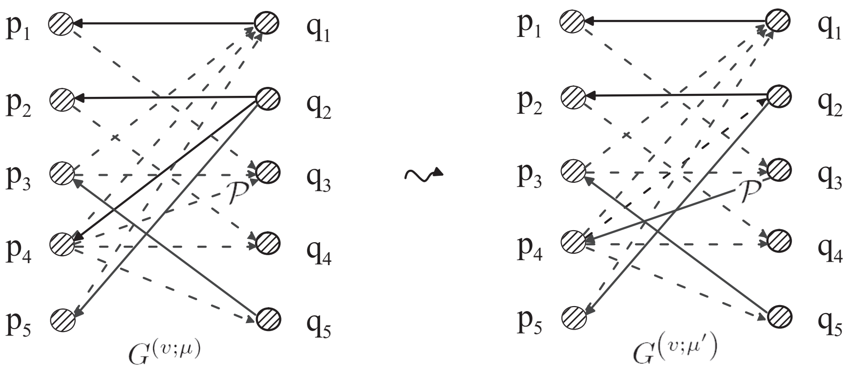

Figure 4.4: Third step in example 4.8.

4. Fourth step:

   We next consider the alternating path $\mathcal{P} = (q_2, p_2, q_4)$ to the unmapped worker 4. In the inner while loop body, we first have to execute the subroutine DisposeRigid(4). This does not change anything, as the worker 4 is unmapped. Afterwards, we alternate the path $\mathcal{P}$. Note that this step increases the size of the mapping image as desired. Finally, the procedure PlaceRigidProposals gets called. However, nothing happens here, since there is trivially no rigidly mapped worker and because there is no unmapped firm at the moment. Summarising, we now have

[20] We could generally adopt the rule of selecting the very first path found in case we have to choose among several paths to ensure the uniqueness of the algorithm. Again, we chose not to include this in the definition of our modified auction algorithm 4.1 for the sake of brevity and the clarity of the exposition.

$\mu = \{1 \to 1, 2 \to 4, 3 \to 5, 4 \to 3, 5 \to 2\}$
$\bar{u} = (6, 5, 6, 5, 6)$
$u = (6, 6, 7, 6, 7)$
$v = (0, 1, 0, 0, 1)$
$Q_R = Q_{2\mu} = Q \backslash \mu(P) = \emptyset$

$D^{(v;\mu)}_{p_1} = \{q_1, q_3\}$
$D^{(v;\mu)}_{p_2} = \{q_2, q_4\}$
$D^{(v;\mu)}_{p_3} = \{q_1, q_3, q_5\}$
$D^{(v;\mu)}_{p_4} = \{q_1, q_2, q_3, q_4, q_5\}$
$D^{(v;\mu)}_{p_5} = \{q_1, q_2\}$

and figure 4.5.

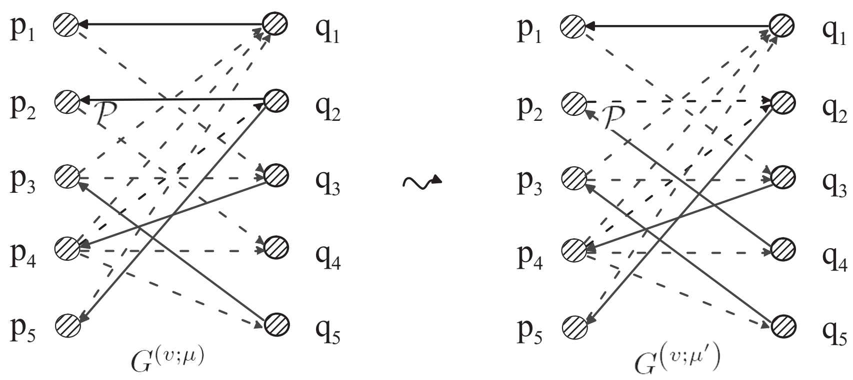

Figure 4.5: Fourth step in example 4.8.

5. Fifth step:

   The condition of the inner while loop is no longer satisfied, since there is no doubly mapped worker anymore and therefore no alternating path at all. Thus, we have to run the procedure HUNGARIANUPDATE now. We trivially obtain $\bar{P} = \bar{Q} = \emptyset$ because $Q_{2\mu} = \emptyset$. We then set $u_1 = 6$, $u_2 = 5$, $u_3 = 6$, $u_4 = 5$ and $u_5 = 6$. This already completes the procedure HUNGARIANUPDATE. Note that the algorithm now leaves the outer while loop because there is no doubly mapped worker anymore. Lastly, line 11 of the main algorithm 4.1 brings no further change. Hence, the algorithm terminates and we have therefore reached a stable outcome $(u, v; \mu)$ with

   $\mu = \{1 \to 1, 2 \to 4, 3 \to 5, 4 \to 3, 5 \to 2\}$
   $\bar{u} = (6, 5, 6, 5, 6)$
   $u = (6, 5, 6, 5, 6)$
   $v = (0, 1, 0, 0, 1)$

The reader should convince himself that the above is really a stable outcome.[21] We note that our modified auction algorithm 4.1 essentially reduces to the famous Hungarian method of Kuhn (1955) in the special case of the assignment game.

[21] See definition 3.5 of a stable payoff and outcome in the DE model or alternatively, the stability definition 3.3 in the assignment game.

Example 4.9

This last example adresses a pure DE market setting that cannot be dealt with in a RB or FB market model. We demonstrate with this example that the auction algorithm of Hochstättler et al. (2006) leads to an endless loop while our modified auction algorithm 4.1 produces a stable outcome as claimed in theorem 4.5.

We consider

$$\begin{aligned} P &= \{p_1, p_2, p_3, p_4, p_5\} \\ Q &= \{q_1, q_2, q_3, q_4, q_5\} \\ R^* &= \left\{ \begin{array}{c} (1,1), (1,3), (1,4), (1,5), (2,2), (2,4), (2,5), \\ (3,1), (3,3), (3,5), (4,2), (4,3), (4,4), (5,1), (5,2) \end{array} \right\} \\ F^* &= \{(1,2), (2,1), (2,3), (3,2), (3,4), (4,1), (4,5), (5,3), (5,4), (5,5)\} \end{aligned}$$

The weight matrices a and b are given below. Recall that we mark the weight entries of rigid and flexible edges in normal and bold face, respectively.

a_{ij}	q_1	q_2	q_3	q_4	q_5
p_1	3	**2**	3	2	4
p_2	**1**	3	**2**	3	2
p_3	4	**2**	4	**1**	4
p_4	**2**	3	4	3	**2**
p_5	4	4	**4**	**1**	**4**

Table 4.4: Weight matrix a in example 4.9.

b_{ij}	q_1	q_2	q_3	q_4	q_5
p_1	3	**2**	3	3	1
p_2	**2**	3	**1**	2	2
p_3	2	**1**	3	**2**	3
p_4	**2**	3	1	2	**2**
p_5	2	3	**3**	**1**	**3**

Table 4.5: Weight matrix b in the example 4.9.

We next show that the above is a pure DE market setting that cannot be dealt with in the RB and the FB market model. Let us first show that the example cannot be described with the RB market model. Note that the flexibility of the edge $(2,1)$ implies that $p_2, q_1 \in F$, since an edge is only flexible in the RB model if both players involved are flexible. The rigidity of the edge $(1,1)$ then implies that p_1 is rigid. This however, means that the edge $(1,2)$ cannot be flexible as we defined it in the current pure DE market example. Finally, we show that we cannot account for the above example within the FB market model. To this end we note that the rigidity of the edge $(1,1)$ implies that $p_1, q_1 \in R$, since an edge is only rigid in the FB model if both players involved are rigid. The flexibility of the edge $(1,2)$ then implies that q_2 is flexible. This however, means that the edge $(2,2)$ cannot be rigid as we defined it in the current pure DE market setting.

1. First step: Identical results for both algorithms

   We set the payoff of all workers at zero and call the procedure PLACERIGIDPROPOSALS.[22] We first compute the potential payoffs of the firms $f_{ij}^{(v;\mu)}$ with (4.1) and then compute the sets $D_i^{(v;\mu)}$ using (4.3). We get

$f_{ij}^{(v;\mu)}$	q_1	q_2	q_3	q_4	q_5
p_1	3	4	3	2	4
p_2	3	3	3	3	2
p_3	4	3	4	3	4
p_4	4	3	4	3	4
p_5	4	4	7	2	7

   and

$$
\begin{aligned}
D_{p_1}^{(v;\mu)} &= \{q_2, q_5\} \\
D_{p_2}^{(v;\mu)} &= \{q_1, q_2, q_3, q_4\} \\
D_{p_3}^{(v;\mu)} &= \{q_1, q_3, q_5\} \\
D_{p_4}^{(v;\mu)} &= \{q_1, q_3, q_5\} \\
D_{p_5}^{(v;\mu)} &= \{q_3, q_5\}
\end{aligned}
$$

   Note that we favour rigid proposals over flexible ones in the procedure PROPOSE(i). Hence, we have the map $\mu = \{1 \to 5, 2 \to 2, 3 \to 1, 4 \to 3, 5 \to 3\}$. We then get $Q_R = \{1, 2, 3, 5\}$. The workers 1, 2 and 5 all have only one proposal, i.e. a rigid proposal. Thus, they do not dispose any firm and we have to set $v_1 = b_{31} = 2$, $v_2 = b_{22} = 3$ and $v_5 = b_{15} = 1$ (always the if-branch). Finally, the best rigid proposal of worker 3 stems from firm 4. Hence, worker 3 chooses firm 4, disposes firm 5 and we set $v_3 = b_{43} = 1$ (if-branch). In the next iteration of the do-while-loop, the unmapped firm 5 again flexibly proposes to worker 3. The best rigid proposal of worker 3 still stems from firm 4 (the only rigid proposal). However, we have already updated the payoff of worker 3 accordingly in the previous iteration. Hence, we now do not execute the if-branch and therefore nothing happens. Note that this is exactly what we want our algorithm to do. A flexible proposal should be kept if it is still optimal for the proposing firm after some increase in the corresponding worker's benefit.[23] The procedure PLACERIGIDPROPOSALS now terminates, as we do not have an unmapped firm at the moment. Summarising, we have

$$
\begin{aligned}
&\mu = \{1 \to 5, 2 \to 2, 3 \to 1, 4 \to 3, 5 \to 3\} && D_{p_1}^{(v;\mu)} = \{q_5\} \\
&\bar{u} = (4, 3, 4, 4, 6) && D_{p_2}^{(v;\mu)} = \{q_2, q_4\} \\
&u = (0, 0, 0, 0, 0) && D_{p_3}^{(v;\mu)} = \{q_1, q_3, q_5\} \\
&v = (2, 3, 1, 0, 1) && D_{p_4}^{(v;\mu)} = \{q_3\} \\
&Q_R = \{1, 2, 3, 5\},\ Q_{2\mu} = \{3\} \text{ and } Q \backslash \mu(P) = \{4\} && D_{p_5}^{(v;\mu)} = \{q_3, q_5\}
\end{aligned}
$$

   and figure 4.6.

[22] For expositional ease, we exploit the procedure PLACERIGIDPROPOSALS of our modified auction algorithm 4.1 in this first step. A corrected version of the procedure PLACERIGIDPROPOSALS of the auction algorithm of Hochstättler et al. (2006) however, would lead to the same results.

[23] Note that we still have $1, 2, 5 \in Q_R$ in this iteration besides the discussed worker $3 \in Q_R$. However, this does not change anything because all these workers only have one proposal and since we have already updated their payoffs in the previous iteration. Thus, we would not carry out the if-branch and therefore nothing would happen. We will skip similar comments in the remainder to ease the exposition.

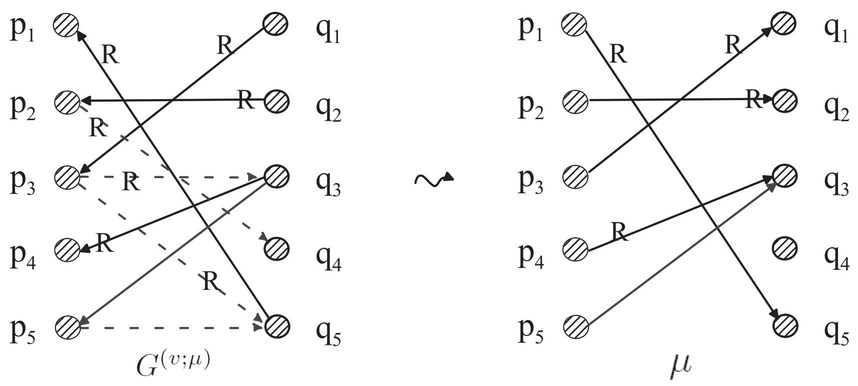

Figure 4.6: First step in example 4.9.

2. Second step with the auction algorithm of Hochstättler et al. (2006):

   The condition of the outer while loop is satisfied with the doubly mapped worker 3. We consider the alternating path $\mathcal{P} = (q_3, p_5, q_5)$ to the worker $5 \in Q_R$, which means that we have to execute the body of the inner while loop. Note that we cannot reach an unmapped worker with an alternating path (no augmenting path) and that worker 5 is the only rigidly mapped worker that can be reached with an alternating path. Moreover, there is no other alternating path to the worker 5. In the auction algorithm of Hochstättler et al. (2006) we first execute the procedure ALTERNATE($\mathcal{P}$). This path alternating however, does not dispose a rigid proposal as desired. Neither does it lead to a worker in Q_{2R}. Afterwards, the procedure PLACERIGIDPROPOSALS gets called. This procedure however, does not change anything. Summarising, we now have

   $\mu = \{1 \to 5, 2 \to 2, 3 \to 1, 4 \to 3, 5 \to 5\}$ $\quad D_{p_1}^{(v;\mu)} = \{q_5\}$
   $\bar{u} = (4, 3, 4, 4, 6)$ $\quad D_{p_2}^{(v;\mu)} = \{q_2, q_4\}$
   $u = (0, 0, 0, 0, 0)$ $\quad D_{p_3}^{(v;\mu)} = \{q_1, q_3, q_5\}$
   $v = (2, 3, 1, 0, 1)$ $\quad D_{p_4}^{(v;\mu)} = \{q_3\}$
   $Q_R = \{1, 2, 3, 5\}$, $Q_{2\mu} = \{5\}$ and $Q \backslash \mu(P) = \{4\}$ $\quad D_{p_5}^{(v;\mu)} = \{q_3, q_5\}$

   and figure 4.7.

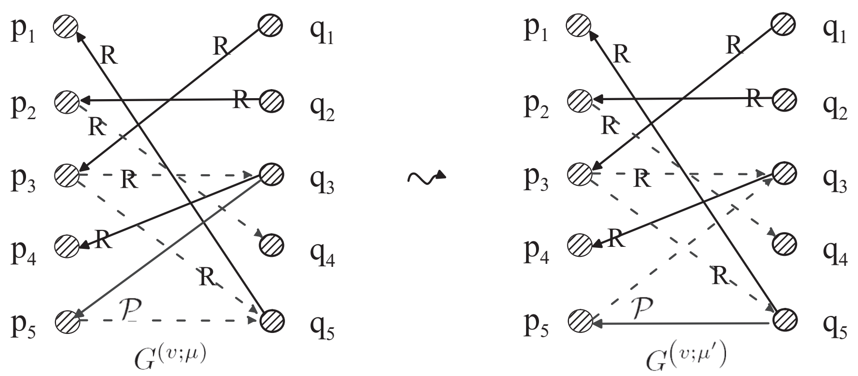

Figure 4.7: Second step with the auction algorithm of Hochstättler et al. (2006) in example 4.9.

3. Third step with the auction algorithm of Hochstättler et al. (2006):

   We consider the alternating path $\mathcal{P} = (q_5, p_5, q_3)$ to the worker $3 \in Q_R$. Note that this worker is the only rigidly mapped worker that can be reached by an alternating path and that the path $\mathcal{P}$ is the only path available. Furthermore, we do not have an alternating path to an unmapped worker (no augmenting path). Thus, we carry out the body of the inner while loop. In the auction algorithm of Hochstättler et al. (2006) we first have to run the procedure ALTERNATE($\mathcal{P}$). Note that this path alternating does not dispose a rigid proposal nor does it lead to a worker in Q_{2R} as desired. Afterwards, the procedure PLACERIGIDPROPOSALS gets called. This procedure however, does not change anything. As can be seen from figure 4.8, we are exactly back to the situation in the beginning of the second step. Moreover, we note that we have never had any choice.[24] Thus, the auction algorithm of Hochstättler et al. (2006) necessarily leads to an endless loop.

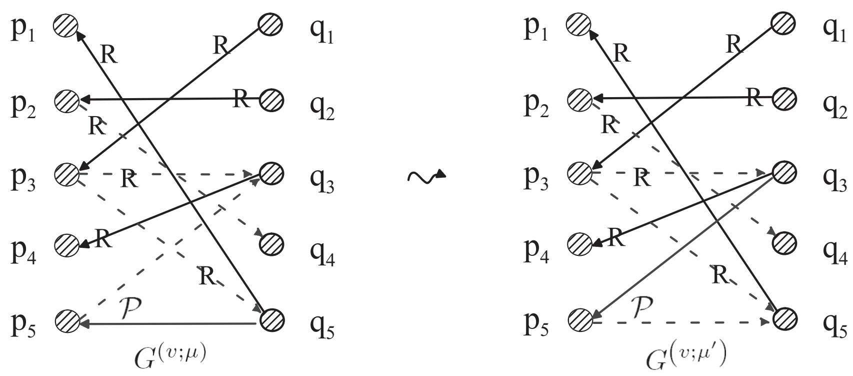

Figure 4.8: Third step with the auction algorithm of Hochstättler et al. (2006) in example 4.9.

Now, we study how our modified auction algorithm 4.1 proceeds after the first step.[25]

2.' Second step with the modified auction algorithm 4.1:

   The condition of the outer while loop is satisfied with the doubly mapped worker 3. We consider the alternating path $\mathcal{P} = (q_3, p_5, q_5)$ to the worker $5 \in Q_R$, which means that we have to execute the body of the inner while loop. Note that we cannot reach an unmapped worker with an alternating path (no augmenting path) and that worker 5 is the only rigidly mapped worker that can be reached with an alternating path. Moreover, there is no other alternating path to the worker 5. As we have shown before the auction algorithm of Hochstättler et al. (2006) would now lead to an endless loop. Let us see how the new auction algorithm 4.1 works at this stage. We first have to execute the additional procedure DISPOSERIGID(5), which removes the rigid proposal $(1, 5)$ as desired. Afterwards, the procedure ALTERNATE($\mathcal{P}$) gets called. This path alternating ensures that the size of the mapping image remains the same as can be seen from figure 4.9.

[24] Strictly speaking, this statement is redundant, as uniqueness is a necessary characteristic of any algorithm. However, one could imagine appropriate rules that pick one option should there be several. From this perspective, it is important that we showed that we never had a choice, since this means that the encountered endless loop in the auction algorithm of Hochstättler et al. (2006) cannot be remedied by any rules.

[25] Again, we could show that the first step leads to the same results with both algorithms. We omit to prove this for the sake of brevity.

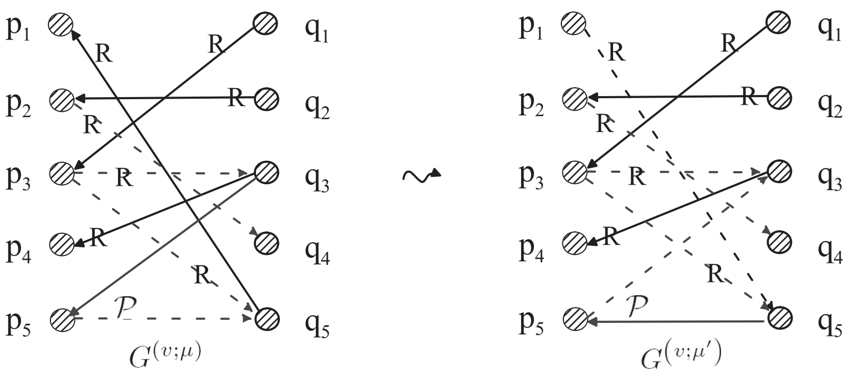

Figure 4.9: Intermediate result in the second step with the modified auction algorithm 4.1 in example 4.9.

Finally, we have to run the procedure PlaceRigidProposals. The currently unmapped firm 1 then rigidly proposes to worker 1. Worker 1 chooses firm 1, disposes firm 3 and we set $v_1 = b_{11} = 3$ (if-branch). In the next iteration the unmapped firm 3 rigidly proposes to worker 3. The best rigid proposal of worker 3 stems from firm 3. Thus, we unmap firm 4 and set $v_3 = b_{33} = 3$ (if-branch). Now, the only unmapped firm 4 rigidly proposes to worker 4. Worker 4 does not have any other proposals. Hence, we do not dispose any proposals and set $v_4 = b_{44} = 2$ (if-branch). This terminates the procedure PlaceRigidProposals, since there are no unmapped firms anymore. Summarising, we now have

$\mu = \{1 \to 1, 2 \to 2, 3 \to 3, 4 \to 4, 5 \to 5\}$	$D_{p_1}^{(v;\mu)} = \{q_1\}$
$\bar{u} = (3,3,4,3,6)$	$D_{p_2}^{(v;\mu)} = \{q_2\}$
$u = (0,0,0,0,0)$	$D_{p_3}^{(v;\mu)} = \{q_3, q_5\}$
$v = (3,3,3,2,1)$	$D_{p_4}^{(v;\mu)} = \{q_4, q_5\}$
$Q_R = \{1,2,3,4\}$, $Q_{2\mu} = Q \backslash \mu(P) = \emptyset$	$D_{p_5}^{(v;\mu)} = \{q_5\}$

and figure 4.10.

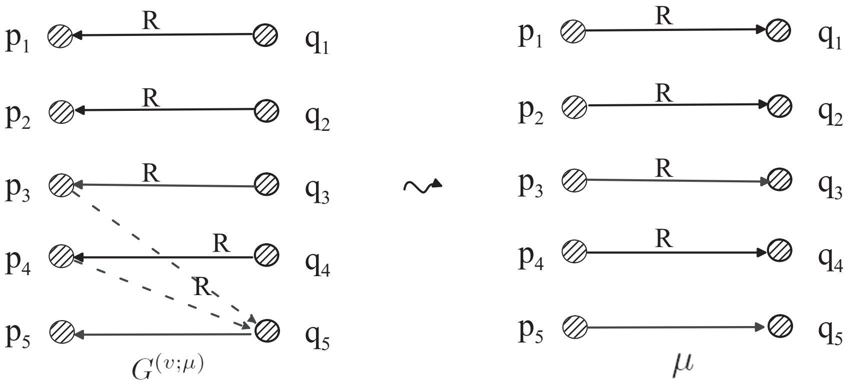

Figure 4.10: Second step with the modified auction algorithm 4.1 in example 4.9.

3.' Third step with the modified auction algorithm 4.1:

Note that we do not have a doubly mapped worker and therefore no alternating path at all. Hence, the condition of the inner while loop is no longer satisfied and we have to carry out the procedure HUNGARIANUPDATE. Because $Q_{2\mu} = \emptyset$ we trivially get $\bar{P} = \bar{Q} = \emptyset$. We set $u_1 = 3$, $u_2 = 3$, $u_3 = 4$, $u_4 = 3$ and $u_5 = 6$. This already completes the procedure HUNGARIANUPDATE. Moreover, we leave the outer while loop because there is no doubly mapped worker anymore. Lastly, line 11 of the main algorithm 4.1 brings no further change. Thus, the algorithm terminates and we have found a stable outcome $(u, v; \mu)$ with

$\mu = \{1 \to 1, 2 \to 2, 3 \to 3, 4 \to 4, 5 \to 5\}$
$\bar{u} = (3, 3, 4, 3, 6)$
$u = (3, 3, 4, 3, 6)$
$v = (3, 3, 3, 2, 1)$

The reader should convince himself that the above is really a stable outcome.[26] Note that the above example adressed a pure DE market setting that cannot be dealt with in a RB or FB market model. It is therefore no surprise that the auction algorithm of Hochstättler et al. (2006) did not work here, as this algorithm is designed for RB markets. The example therefore showed the necessity of modifying the auction algorithm of Hochstättler et al. (2006). Finally, this last example demonstrated that our modified auction algorithm 4.1 produced a stable outcome as claimed in theorem 4.5.

[26] See definition 3.5 of a stable payoff and outcome in the DE model.

Bibliography

R. Ahuja, T. Magnanti, and J. Orlin. *Network Flows.* Prentice Hall, 1993.

M. Aigner. *Diskrete Mathematik.* Vieweg Verlag, Wiesbaden, sixth edition, 2006.

G. B. Dantzig. *Linear Programming and Extensions.* Princeton University Press, Princeton, 1963.

K. Eriksson and J. Karlander. Stable Matching in a Common Generalization of the Marriage and Assignment Models. *Discrete Mathematics*, 217(1):135–156, 2000.

A. Frank. On Kuhn's Hungarian Method - A Tribute from Hungary. Technical report, Egerváry Research Group on Combinatorial Optimization, October 2004.

S. Fujishige and A. Tamura. A Two-Sided Discrete-Concave Market with Bounded Side Payments: An Approach by Discrete Convex Analysis. RIMS Preprint No. 1470, Kyoto University, August 2004.

D. Gale and L. S. Shapley. College Admissions and the Stability of Marriage. *American Mathematical Monthly*, 69:9–15, 1962.

Z. Galil. Efficient Algorithms for Finding Maximum Matching in Graphs. *ACM Computing Surveys*, 18(1):23–28, 1986.

W. Hochstättler. Kombinatorische Optimierung. Vorlesung WS 1999/2000. Found on the 5th of June 2007 on `http://www.math.tu-cottbus.de/INSTITUT/lsgdi/lsvorlWS1999.html`, 1999.

W. Hochstättler, H. Jin, and R. Nickel. The Hungarian Method in a Mixed Matching Market. Technical report, FernUniversität in Hagen, 2005.

W. Hochstättler, H. Jin, and R. Nickel. Note on an Auction Procedure for a Matching Game in Polynomial Time. pages 387–394. AAIM Proceedings, 2006.

H. Jin. Algorithmen für Matching-Märkte. Master's thesis, Technische Universität Cottbus, November 2005.

D. Jungnickel. *Graphs, Networks and Algorithms.* Springer, Berlin, Heidelberg, New York, 2nd edition, 2005.

H. Kuhn. The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quaterly*, 2:83–97, 1955.

A. E. Roth. The Evolution of the Labor Market for Medical Interns and Residents: A Case Study in Game Theory. *Journal of Political Economy*, 92:991–1016, 1984.

A. E. Roth. A Natural Experiment in the Organization of Enty Level Labor Markets: Regional Markets for New Physicians and Surgeons in the U.K. *American Economic Review*, 81:415–440, 1991.

A. E. Roth and M. Sotomayor. Stable Outcomes in Discrete and Continuous Models of Two-Sided Matching: A Unified Treatment. *Revista de Econometrica, The Brazilian Review of Econometrics*, 16(2), November 1996.

A. E. Roth and M. Sotomayor. *Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis*. Cambridge University Press, Cambridge, 1999.

A. E. Roth and J. H. Vande Vate. Random Paths to Stability in Two-Sided Matching. *Econometrica*, 58(2):1475–1480, 1990.

L. S. Shapley and M. Shubik. The Assignment Game I: The Core. *International Journal of Game Theory*, 1:111–130, 1972.

M. Sotomayor. Existence of Stable Outcomes and the Lattice Property for a Unified Matching Market. *Mathematical Social Sciences*, 39(2):119–132, 2000.

Ich bestätige hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

St. Gallen, August 2007 David Schiess

Curriculum vitae

David Schiess, born on the 18th of March 1976 in Münsterlingen as son of Ernst and Renate Schiess; two brothers Raphael and Josua.

Education

2003 -	Studies in Mathematics and Computer Sciences, Bachelor, FernUniversität Hagen, Germany
2002 - 2007	Doctoral Studies in Economics, University of St. Gallen (HSG)
2002 - 2003	Swiss Program for Doctoral Students in Economics, Studienzentrum Gerzensee, Foundation of the Swiss National Bank
1997 - 2001	Studies in Economics (lic. oec.), University of St. Gallen (HSG)
1991 - 1996	Matura Typus B, Kantonsschule Kreuzlingen (TG)

Work Experience

2007 -	Asset and Liability Management, Programmer and Pension Analyst, c-alm AG, St. Gallen (SG)
2006 -	Freelance Mathematical Programmer and Consultant, Adwise GmbH, St. Gallen (SG)
2006 -	Wealth and Risk, Research “Schwerpunkt” in Finance, Banking and Insurance, University of St. Gallen (HSG)
2001 -	Teaching Assistant, Department of Mathematics and Statistics, University of St. Gallen (HSG)
2000 -	Research Assistant, Department of Mathematics and Statistics, University of St. Gallen (HSG)
2000	Research Assistant, Department of Economics, University of St. Gallen (HSG)
1996 - 1997	Sales and Logistics, Blaser Getränke AG, Kreuzlingen (TG)