The Hungarian Method in a Mixed Matching Market

Winfried Hochstättler
Department of Mathematics
FernUniversität in Hagen
D-58084 Hagen

Hui Jin
Department of Mathematics
BTU Cottbus
D-03013 Cottbus

Robert Nickel
Department of Mathematics
FernUniversität in Hagen
D-58084 Hagen

October 17, 2005

Abstract

We present an algorithm that computes a stable matching in a common generalization of the marriage and the assignment game in $\mathcal{O}(n^4)$ time.

1 Introduction

Since its introduction by Gale and Shapley [8] the stable marriage problem has become quite popular among scientists from different fields such as game theory, economics, computer science, and combinatorial optimization. Among others this is mirrored by three monographs: Knuth [12], Gusfield and Irving [10], Roth and Sotomayor [15]. The problem is the following: Given two disjoint groups of players (men-women or workers-firms etc.), where each player is endowed with a preference list on the other group, the objective is to match the players from one group to players from the other group such that there is no pair which is not matched but prefers each other over their partners. Gale and Shapley [8] showed that such a *stable matching* always exists. The proof is algorithmic and the algorithm has become famous under the name "men propose – women dispose".

In their award winning book Roth and Sotomayor [15] observed that the set of stable solutions from another game on bipartite matching, namely the assignment game [16], has several structural similarities with the set of stable matchings. They challenged the readers to find a unifying theory for the two games. In the assignment game we are given a weighted bipartite graph. A solution consists of a matching and an allocation of its weight to the players.

A solution is stable if no pair gets allocated less than the weight of its connecting edge. Shapley and Shubik [16] observed that this condition is identical to the dual constraints of the linear programming model for weighted bipartite matching, thus the dual variables in an optimal solution coincide with the stable allocations. However, algorithms and complexity issues of game theoretic solution concepts have raised attention only recently (see e. g. Deng and Papadimitriou [2], Faigle et al. [5], Deng et al. [3]) and the classical algorithm for weighted bipartite matching, namely the Hungarian Method of Kuhn [13], is not as prominent in game theory as it is in combinatorial optimization. However, Demange et al. [1] claim that their "exact" auction procedure that proves the existence of stable solutions were a variant of Kuhn's method.

Roth and Sotomayor [14] themselves presented a first model unifying stable matching and the assignment game and showed that its set of stable solutions, if it is non-empty, has the desired structural properties. Eriksson and Karlander [4] modified this model to the one presented in this paper and gave an algorithmic proof of the existence of a stable solution. For the classical special cases, their algorithm coincides with "men propose – women dispose", respectively with the "exact" auction procedure of [1]. As implemented, this algorithm is not polynomial time but pseudopolynomial. Also it yields a proof of the existence of stable solutions in presence of irrational data only via arguments from nonstandard analysis. A careful analysis [11] of the algorithm, though, reveals that a proper implementation solves the problem in $\mathcal{O}(n^4)$ similar to the algorithm presented here.

Sotomayor [17] gave an alternative proof of the existence of stable solutions of the model of Eriksson and Karlander. Fujishige and Tamura [7] called that proof non-constructive. Dissenting from that, the purpose of the present paper is to extract an algorithm from the key lemma of Sotomayor [17] that computes a stable solution in $\mathcal{O}(n^4)$. In the case of the assignment game this algorithm specializes to the implementation of the Hungarian Method where a search tree (and not a forest) is grown starting from a single unmatched vertex. For stable marriage we derive a sort of asynchronous implementation of "men propose – women dispose" that does not proceed in rounds.

In the next section we introduce the model, discuss its special cases in Section 3 and present and analyze our algorithm in the last section. We assume some familiarity with bipartite matching and combinatorial optimization. Our notation should be fairly standard.

2 Notation

The following model, originally introduced by Eriksson and Karlander [4], displays a two-sided market, where we have two types of players, P and Q which we call *firms* and *workers*. Moreover, both sets are again subdivided into *flexible* (F) and *rigid* players (R) so that $P \dot{\cup} Q = F \dot{\cup} R$. If a firm $i \in P$ is matched to a worker $j \in Q$ they get a certain benefit $a_{ij} + b_{ij}$ from that relationship. If both players are flexible, they can negotiate on how to split up this amount. If at least one player is rigid i receives a_{ij} and j receives b_{ij}.

Thus, mathematically we have the following. Let $G = (P \cup Q, E, a, b)$ be a complete bipartite graph with two non-negative weight functions $a, b : E \to \mathbb{R}_+$ and $P \dot{\cup} Q = F \dot{\cup} R$ another partition of the vertices.

A set $M \subseteq E$ is called a *matching* if each vertex of G is contained in at most one edge of M and we denote by $V_M \subseteq V$ the set of matched vertices of G. A pair of functions $u : P \to \mathbb{R}$ and $v : Q \to \mathbb{R}$ is called a *payoff.* An *outcome* of the game is a triple $(u, v; M)$ consisting of a payoff and a matching. Such an outcome $(u, v; M)$ is called *feasible* if

(i) $u_i \geq 0$ and $v_j \geq 0$ for all $(i, j) \in P \times Q$ and $u_i = 0$ (resp. $v_j = 0$) if i (resp. j) is unmatched.

(ii) $u_i + v_j = a_{ij} + b_{ij}$ for $(i, j) \in M$ and $\{i, j\} \subseteq F$.

(iii) $u_i = a_{ij}$ and $v_j = b_{ij}$ for $(i, j) \in M$ and $\{i, j\} \cap R \neq \emptyset$.

Accordingly, we call an edge (i, j) *flexible* if $\{i, j\} \subseteq F$, *rigid* otherwise, and denote by F^* resp. R^* the set of flexible resp. rigid edges.

Now, we can define our notion of stability:

Definition 2.1. *A payoff* (u, v) *is called* stable *if for all* $(i, j) \in P \times Q$ *we have that*

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

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

An outcome $(u, v; M)$ *is called* stable *if it is feasible and* (u, v) *is a stable payoff.*

Note that this notion of stability coincides with Eriksson and Karlander [4] and [17] only for outcomes. A pair $(i, j) \in P \times Q$ that violates one of (i) and (ii) is called a *blocking pair.* For any blocking pair (i, j) we define

$$u_i^j := \begin{cases} a_{ij} + b_{ij} - v_j & \text{if } (i, j) \in F^* \\ a_{ij} & \text{if } (i, j) \in R^*. \end{cases}$$

A blocking partner j that maximizes u_i^j is called *i's favorite blocking partner.*

3 Special Cases

If one of F or R is empty the model from the last section reduces to the well-known *stable marriage* resp. *assignment game.* In this section we will discuss these models and recall an algorithm for each of them that we will merge into a single algorithm for the mixed model in Section 4.

Stable Marriage

If $F = \emptyset$ the numbers a_{ij} (for the firms) and the numbers b_{ij} (for the workers) induce a preference list for each firm resp. worker. A matching M is stable if no non-matching edge $(i, j) \notin M$ has the property that i prefers j to its matching partner as well as j prefers i to its matching partner, i. e. $u_i < a_{ij}$ and $v_i < b_{ij}$. The famous algorithm to find a stable matching is due to Gale and Shapley [8]. We need a slightly modified version (see Algorithm 1) which differs from the known "men propose – women dispose" algorithm in a way that proposals are not made in rounds but asynchronously.

Algorithm 1 Asynchronous "men propose – women dispose"

```
while ∃ an unmatched firm $i$ do
    $i$ asks favorite $j$ to join
    if $j$ prefers $i$ over its current partner $i_0$ then
        $j$ deletes $i_0$ from preference list
        UNMATCH($j$)
        MATCH($i$,$j$)
    else
        $i$ deletes $j$ from preference list
    end if
end while
```

Assignment Game

If $R = \emptyset$ the problem reduces to the assignment game or weighted bipartite matching and its dual linear program. The payoffs are the dual variables (i. e. a weighted vertex cover), the stability condition reduces to their feasibility, and a maximum matching together with a minimum weighted vertex cover yield a stable outcome by linear programming duality.

A famous algorithm to find a maximum weighted matching and a minimum weighted vertex cover in $\mathcal{O}(n^3)$ is Kuhn's Hungarian Method (see Algorithm 2 or e. g. Frank [6] for a transparent presentation). It starts with a weighted vertex cover (resp. a dually feasible i. e. stable payoff). For a given bipartite graph $G = (P \dot{\cup} Q, E)$, a matching M, and a payoff (u, v) the *digraph of tight edges* $G_{(u,v;M)}$ is defined as the bipartite digraph with all vertices of G, forward edges $(i, j) \notin M$ that satisfy $u_i + v_j = a_{ij} + b_{ij}$, and backward edges (j, i) for $(i, j) \in M$. An *augmenting path* in $G_{(u,v;M)}$ is a directed path that starts with an unmatched firm and ends with an unmatched worker. ALTERNATE($\mathcal{P}$) interchanges matching and non-matching edges on the alternating path $\mathcal{P}$. For a vertex $k \in P \times Q$ the function BFS(k) returns all vertices $\bar{P} \dot{\cup} \bar{Q}$ from $P \cup Q$ that are connected to k by a directed path in the digraph of tight edges.

Algorithm 2 Modified Hungarian Method

```
procedure WeightedBipartiteMatching
    for all $i \in P$ do
        $u_i \leftarrow \max(\{a_{ij} + b_{ij} \mid j \in Q\})$
    end for
    while $(u, v; M)$ is not stable do
        if $\exists$ augmenting path $\mathcal{P}$ in $G_{(u,v;M)}$ then
            alternate($\mathcal{P}$)
        else
            $i \leftarrow$ unmatched firm
            HungarianUpdate($i$)
        end if
    end while
end procedure

procedure HungarianUpdate($i$)
    $\bar{P} \dot{\cup} \bar{Q} \leftarrow$BFS($G_{(u,v;M)}$,$i$)
    $\Delta \leftarrow \min\{u_i + v_j - a_{ij} - b_{ij} \mid i \in \bar{P}, j \notin \bar{Q}\} > 0$
    for all $i \in \bar{P}$ do
        $u_i \leftarrow u_i - \Delta$
    end for
    for all $j \in \bar{Q}$ do
        $v_j \leftarrow v_j + \Delta$
    end for
end procedure
```

4 An Algorithm to Find a Stable Outcome

Sotomayor [17] has shown that there is always a stable outcome. The ingredients of her proofs provided ideas for our algorithm that makes use of a modified hungarian update and augmenting path techniques.

By eventually introducing dummy firms or workers we may assume that $|P| = |Q| =: n$. Such a dummy k satisfies $a_{kj} = 0$ for all workers resp. $b_{ik} = 0$ for all firms.

For a given outcome $(u, v; M)$ we define an *augmentation graph* $G_{(u,v;M)}$ as a subgraph of G with edge set

$$E_{(u,v;M)} := \{(j, i) \mid (i, j) \in M\} \cup \{(i, j) \mid j \in D_i(u, v; M)\},$$

where

$$\begin{aligned} D_i(u,v; M) := \{&i\text{'s favorite blocking partners}\} \\ &\cup \{j \in Q \mid (i, j) \in F^* \setminus M \text{ and } u_i + v_j = a_{ij} + b_{ij}\} \\ &\cup \{j \in Q \mid (i, j) \in R^* \setminus M \text{ and } u_i = a_{ij} \text{ and } v_j < b_{ij}\}. \end{aligned}$$

The following path augmentation argument is a slight modification of Sotomayor [17, Lemma 1].

Lemma 4.1. *Let $(u, v; M)$ be a feasible outcome such that no matched firm is contained in a blocking pair. If there is an unmatched firm i_1 with a blocking partner then there exists a feasible outcome $(\tilde{u}, \tilde{v}; M)$ and an oriented path $\mathcal{P}$ in $G_{(\tilde{u},\tilde{v};M)}$ starting from i_1 that reaches a player of R, an unmatched worker or a firm with payoff zero.*

Proof. Let $\bar{P} \subseteq P$ and $\bar{Q} \subseteq Q$ be the firms and workers reachable from i_1 in $G_{(u,v;M)}$ and assume there is no such path. Then $(\bar{P} \cup \bar{Q}) \cap R = \emptyset$, $V_M \cap \bar{Q} = \bar{Q}$ and $u_i > 0 \quad \forall i \in \bar{P}$. If $i \in \bar{P} \setminus \{i_1\}$ then i is matched and by assumption is not contained in any blocking pair, i. e. for all $j \in Q$ $u_i \geq a_{ij} + b_{ij} - v_j$ and if $j \in R$ then either $u_i \geq a_{ij}$ or $v_j \geq b_{ij}$. For each edge $(i, j) \in (\bar{P} \setminus \{i_1\}) \times (Q \setminus \bar{Q})$ it, thus, follows from the definition of $G_{(u,v;M)}$ that $u_i > a_{ij} + b_{ij} - v_j$ and if $j \in R$ and $v_j < b_{ij}$ then $u_i > a_{ij}$. Now let

$$\begin{aligned}
&\tilde{F} := (\bar{P} \times (Q \setminus \bar{Q})) \cap F^* \text{ and} \\
&\tilde{R} := \{(i, j) \in \bar{P} \times ((Q \setminus \bar{Q}) \cap R) \mid u_i > a_{ij} \text{ and } v_j < b_{ij}\}.
\end{aligned}$$

We modify the outcome such that it remains feasible until we either get a new edge in $G_{(u,v;M)}$ or a firm with payoff zero, i. e. let

$$\begin{aligned}
\Delta := \max\{\delta \mid & u_i - \delta \geq 0 && \forall i \in \bar{P} && (1)\\
& u_i - \delta \geq a_{ij} && \forall (i, j) \in \tilde{R} && (2)\\
& u_i + v_j \geq a_{ij} + b_{ij} && \forall (i, j) \in \tilde{F}\}. && (3)
\end{aligned}$$

By the above then $\Delta > 0$. We construct a new outcome $(\tilde{u}, \tilde{v}; M)$ by decreasing u_i and increasing v_j by Δ for all $(i, j) \in (\bar{P} \setminus \{i_1\}) \times \bar{Q}$. By construction, this outcome is still feasible. Δ has been chosen such that equality holds in one of (1-3) for at least one edge (i, j). If (1) holds with equality then there is a firm with payoff zero in $\bar{P}$ and if (2) holds with equality we reach a worker in R. Otherwise, we reach another flexible worker from $Q \setminus \bar{Q}$ and enlarge $\bar{Q}$. If the assumption of the lemma is still satisfied, i.e. there is still no such path, we iterate the process. The latter can happen at most $|Q|$ times. Thus the process is finite and eventually ends with a path since there must be at least one unmatched worker as $|P| = |Q|$. □

The modification of the payoffs in the above proof is similar to the update of the dual variables in HungarianUpdate. A major difference is that the Hungarian Method always ensures dually feasible variables, namely $u_i + v_j \geq a_{ij} + b_{ij} \quad \forall (i, j) \in P \times Q$. In our terms this corresponds to a stable payoff. We adapt this idea and introduce *virtual payoffs* denoted by $\bar{u}_i$ for all firms such that no edge (i, j) ever forms a blocking pair in the (infeasible) outcome $(\bar{u}, v; M)$ during the algorithm. Thus u_i is the allocation that we can afford from the current matching and $\bar{u}_i$ is an upper bound on the highest possible benefit of firm i from the current market situation. Our approach thus becomes similar to complementarity algorithms known from combinatorial optimization: We have a (primal) feasible matching M and a stable (dually feasible) payoff $(\bar{u}, v)$. Optimality is reached when $\bar{u}_i > 0$ implies that u_i is matched for all i.

The algorithm then can be outlined as follows: We start with an empty matching, payoff zero for all workers, and the maximum possible individual payoff for each firm. Throughout the algorithm we ensure that no firm that has a matching partner belongs to a blocking pair. As long as there are blocking pairs, we consider the connected component of the corresponding unmatched firm i formed by D_i and construct a path as in Lemma 4.1. For that purpose we, eventually modify the outcomes. Once such a path is found we either increment the matching, discard a firm, or discard an edge from R^*.

We first adjust HungarianUpdate choosing Δ as in (1-3). Furthermore, in line 15 of HungarianUpdate we modify $\bar{u}$ for all vertices in $\bar{P}$ and u for all vertices in $\bar{P} \setminus \{i_1\}$. The algorithm (see Algorithm 3) starts with a feasible outcome $(u, v; M) \leftarrow (0, 0; \emptyset)$ and a stable payoff $(\bar{u}, v)$:

$$\bar{u}_i \leftarrow \max\left(\{a_{ij} + b_{ij} \mid (i,j) \in F^*\} \cup \{a_{ij} \mid (i,j) \in R^*\}\right) \quad i \in P.$$

Algorithm 3 Construction of a Stable Outcome

```
while blocking pair $(i_1, j)$ exists do
    $j_1 := i_1$'s favorite blocking partner
    while there is no path $\mathcal{P}$ as in Lemma 4.1 do
        HungarianUpdate($j_1$)
    end while
    PathUpdate($\mathcal{P}$)
end while
```

In the next section we will discuss how we implement the update of the outcome according to $\mathcal{P}$.

4.1 The Augmentation Step

The details of the path update procedure are worked out in Algorithm 4 where we make use of the following sub-procedures.

Alternate gets an alternating path (resp. an alternating cycle in CASE 3.3) $\mathcal{P} = (i_1, j_1, i_2, j_2, \ldots)$ as argument, i. e. every second edge is a matching edge. Matching and non-matching edges are exchanged along the path (resp. cycle) such that former matching edges become non-matching edges and vice versa. Hence, the number of matching edges does not change in case $\mathcal{P}$ starts in an unmatched firm and ends in a firm or if $\mathcal{P}$ is a cycle and increases by 1 if it starts in an unmatched firm and ends in an unmatched worker. Other cases will not occur. If $\mathcal{P} = (i_1, j_1, i_2, j_2, \ldots)$ is the path as in Lemma 4.1 then $\mathcal{P}_{[i_1, j_{s-1}]}$ denotes the subpath from i_1 to j_{s-1} and if $i_k \in R$ is matched to $j_k = j_s$ with $1 \leq k < s$ then $\mathcal{P}_{[j_k, j_s]}$ denotes the alternating cycle composed of the subpath of $\mathcal{P}$ from j_k to i_s and the matching edge (j_k, i_k).

Algorithm 4 Augmentation along the path $\mathcal{P}$

```
procedure PathUpdate($\mathcal{P}$)
    if $\mathcal{P} = (\overbrace{i_1, j_1, \ldots, i_s}^{F}, j_s)$ and $j_s$ unmatched then          ▷ CASE 1
        Alternate($\mathcal{P}$)
        Update($i_1, j_1$)
        Update($i_s, j_s$)
    else if $\mathcal{P} = (\overbrace{i_1, j_1, \ldots, i_{s-1}, j_{s-1}, i_s}^{F})$, $u_{i_s} = 0$ then
                                                                            ▷ CASE 2
        Alternate($\mathcal{P}$)
        Update($i_1, j_1$)
        Discard($i_s$)
    else if $\mathcal{P} = (\overset{R^*}{i_1}, j_1)$ then                                ▷ CASE 3.1
        Unmatch($j_1$)
        Alternate($(i_1, j_1)$)
        Update($i_1, j_1$)
    else if $\mathcal{P} = (\overbrace{i_1, j_1, \ldots, i_{s-1}, j_{s-1}}^{F}, \overset{R^*}{i_s}, j_s)$ then          ▷ CASE 3.2
        Unmatch($j_s$)
        Alternate($\mathcal{P}$)
        Update($i_1, j_1$)
        Update($i_s, j_s$)
    else if $\mathcal{P} = (\overbrace{i_1, j_1, \ldots, i_k, j_k = j_s, \ldots}^{F}, \overset{R}{i_s}, \overset{F}{j_s})$ then     ▷ CASE 3.3
        Alternate($\mathcal{P}_{[j_k, j_s]}$)
        Update($i_s, j_s$)
    else if $\mathcal{P} = (\overbrace{i_1, j_1, \ldots}^{F}, \overset{R}{i_s})$ and $D_{i_s} = \emptyset$ then           ▷ CASE 3.4
        Unmatch($j_{s-1}$)
        Alternate($\mathcal{P}_{[i_1, j_{s-1}]}$)
        Update($i_1, j_1$)
        if $i_s$ has blocking partner then
            $j_s \leftarrow$ favorite blocking partner of $i_s$
            PathUpdate($(i_s, j_s)$)
        else
            Discard($i_s$)
        end if
    end if
end procedure
```

UPDATE(i,j) sets $\bar{u}_i \leftarrow u_i \leftarrow a_{ij} + b_{ij} - v_j$ in case $(i,j) \in F^*$ and $\bar{u}_i \leftarrow u_i \leftarrow a_{ij}, v_j \leftarrow b_{ij}$ if $(i,j) \in R^*$.

UNMATCH(j) removes a matching edge (i,j), and sets $u_i \leftarrow 0$.

DISCARD(i) sets $\bar{u}_i \leftarrow u_i \leftarrow 0$. Such a firm will never ever form a blocking pair.

4.2 Correctness and Complexity

We are now going to prove that Algorithm 3 is correct. This directly follows from some invariants of the algorithm.

Lemma 4.2. *After each call of* HUNGARIANUPDATE *or* PATHUPDATE *the following holds:*

a) $(u,v;M)$ *is feasible.*

b) (i,j) *blocks* $(u,v;M) \Rightarrow i$ *is unmatched.*

c) $u_i = \bar{u}_i \iff i$ *is matched or* $u_i = \bar{u}_i = 0$.

d) $(\bar{u}, v)$ *is stable.*

e) For all $i \in P$ $\bar{u}_i$ *did not increase.*

f) For all $j \in Q$ v_j *did not decrease .*

g) $|M|$ *did not decrease.*

h) Once a firm has been discarded it will never be matched again.

Proof. In the first step of the algorithm all conditions hold as we start with a sufficiently large virtual payoff $\bar{u}_i$ and an outcome $(0,0;\emptyset)$. Now assume conditions a) to h) are true for the current step of the algorithm. We show that this is still true after the next call of the procedures considered.

f) We start with $v = 0$ and v is altered only in HUNGARIANUPDATE, when the matching is augmented or when an edge in R^* is matched (CASES 3.1, 3.2 and 3.3). HUNGARIANUPDATE only increases some v_j and an unmatched worker had a payoff of zero before. An edge in $(i,j) \in R^*$ can be matched only if either (i,j) forms a blocking pair or $u_i = a_{ij}$ and $v_j < b_{ij}$. In both cases $v_j < b_{ij}$ holds and UPDATE(i,j) increases v_j.

g) The matching is altered in PATHUPDATE only. In CASE 1 it is increased. In all other cases we either alternate on an even path or an even cycle or unmatch a vertex and immediately augment the matching again. Thus in all cases the size of the matching is not changed.

h) A firm i is discarded in CASE 2, if it was matched, and thus had no blocking partner by b), with $u_i = \bar{u}_i = 0$ and has become unmatched. Or it has become unmatched, enforcing $u_i = 0$ and has no blocking partner in CASE 3.4. In that case we can set $\bar{u}_i = 0$ and remain dually feasible. By f) and as a_{ij} and b_{ij} are non-negative i will never have a blocking partner again. After the algorithm has terminated we may match such a firm to some unmatched v_j, necessarily satisfying $a_{ij} = 0$.

c) When an unmatched firm is matched, it is matched to its favorite blocking partner, updated and $u_i = \bar{u}_i$ holds. When a firm becomes unmatched its payoff is set to zero.

d),e) We consider situations where an outcome is updated. In HungarianUpdate the definition of Δ ensures that d) is not violated. Furthermore, $\bar{u}_i$ only decreases for some firms. In PathUpdate $\bar{u}$ is altered when Update(i, j) or Discard(i_s) is called. The latter case has been discussed before and does not cause blocking pairs. When Update(i, j) is invoked, then $(i, j) \in D_i(u, v; M)$, in particular $(i, j) \notin M$ and we have the following cases:

i was already matched (CASES 1, 3.2, 3.3, 3.4) and
$(i, j) \in F^* \Rightarrow$Update$(i, j)$ has no effect as v_j has not been changed
$(i, j) \in R^* \Rightarrow$ by definition of D_i $\bar{u}_i = u_i$ does not change while v_j increases

j is i's favorite blocking partner (CASES 1, 2, 3.1, 3.2, 3.4) and
$(i, j) \in F^* \Rightarrow \bar{u}_i \leftarrow \max\{a_{ij} + b_{ij} - v_j \mid (i, j) \text{ forms a blocking pair}\}$
$(i, j) \in R^* \Rightarrow \bar{u}_i \leftarrow \max\{a_{ij} \mid (i, j) \text{ forms a blocking pair}\}$

Hence, d) holds as in any case $\bar{u}_i + v_j \geq a_{ij} + b_{ij}$ holds for all edges in F^* and $\bar{u}_i \geq a_{ij}$ for all edges in R^*. As $\bar{u}_i$ changes at most if i was unmatched, e) follows from the fact that $\bar{u}_i$ was dually feasible before the procedure call and is now changed to the minimal value guaranteeing dual feasibility.

Finally we discuss a) and b):

HungarianUpdate: The procedure modifies outcomes only in $\bar{P} \cup \bar{Q}$. Note that any firm in $\bar{P} \setminus \{i_1\}$ is matched and therefore $u_i = \bar{u}_i \quad \forall i \in \bar{P} \setminus \{i_1\}$. Since for a matching edge $(i, j) \in M$ u_i and v_j are modified in opposite direction the edge (i, j) remains tight. Furthermore, u is decreased at most only until the first i satisfies $u_i = 0$, thus $(u, v; M)$ remains feasible. In $\bar{P} \cup \bar{Q}$ $u_i + v_j$ is not altered for any edge and thus we do not have any blocking pair. By partially increasing v workers in $\bar{Q}$ become less attractive for firms outside $\bar{P}$. It follows that no new blocking pairs occur and b) holds.

CASE 1/2: All edges along $\mathcal{P}$ are tight except for the first edge which is made tight by Update. Hence, all newly matched edges are tight. Before (i, j)

is updated we have

$$0 = u_{i_1} < a_{i_1 j_1} + b_{i_1 j_1} - v_{j_1} \leq \bar{u}_{i_1}$$

and therefore $u_{i_1} > 0$ after the update and the outcome is feasible. There are no new blocking pairs since v is not changed, i_1 is matched to its favorite blocking partner and there is no other newly matched firm. In CASE 2 a firm is discarded but actually this step does nothing as has been discussed before.

CASE 3.1: The feasibility is obvious and there are no newly matched firms instead of i_1 which is matched to its favorite blocking partner.

CASE 3.2: Again, the new outcome is still feasible. i_1 has no blocking partner after it is matched to its favorite. Since $(i_s, j_s) \in R^*$ we have $u_{i_s} = a_{i_s j_s}$ and $v_{j_s} < b_{i_s j_s}$. Hence, an update does not change the outcome of i_s and makes j_s less attractive to other firms.

CASE 3.3: Only the outcome of j_s is modified in a direction that makes j_s less attractive. No new blocking pairs are formed and feasibility is not violated.

CASE 3.4: If i_s does not have any new blocking partner then $v_j \geq a_{i_s j} + b_{i_s j}$ for all $j \in Q$ and we can set $\bar{u}_{i_s} \leftarrow 0$ without violating d). Therefore a) and b) are obviously still true. Otherwise, i_s is matched to its favorite blocking partner.

□

In each of the CASES 3.1 – 3.4 an edge from R^* is made tight. By Lemma 4.2 and definition of D_i such an edge will never appear as non-matching edge in any search tree again. Therefore, we may say it is *discarded.* Correctness and a complexity result now follow from the immediate observation that:

Proposition 4.3. *After every call of* PathUpdate *one of the following statements holds:*

1. *$|M|$ has been increased.*
2. *A firm has been discarded.*
3. *An edge from R^* has been discarded.*

Theorem 4.4. *Algorithm 3 computes a stable outcome and can be implemented to run in $\mathcal{O}(n^4)$.*

Proof. From Lemma 4.1 we conclude that as long as there is some blocking pair there also must be a path $\mathcal{P}$. The matching can be augmented at most $|Q|$ times, there are only $|P|$ firms to discard and at most $|P| \cdot |Q|$ edges to be discarded in R^*. The correctness thus follows from Lemma 4.2 and Proposition 4.3.

There is not much work left to derive an implementation that runs in $\mathcal{O}(n^4)$. By Proposition 4.3 all that is left to show is that we can implement the **while**-loop in line 3 of Algorithm 3 to be passed in $\mathcal{O}(n^2)$. For that purpose we use a standard implementation of the *Hungarian Method* see e. g. Galil [9].

For any fixed $j \in Q$ we store a value Δ_j corresponding to Δ in (1-3) being the *distance* from j to $\bar{P} \cup \bar{Q}$. Δ_j must be updated for each j in $\mathcal{O}(n)$ whenever we add a vertex to $\bar{P} \cup \bar{Q}$ (which happens $\mathcal{O}(n)$ times). For computing $\Delta = \min\{\Delta_j > 0\}$ we freely may also spend $\mathcal{O}(n)$ time. After an update of the payoffs we can set $\Delta_j := \max\{0, \Delta_j - \Delta\}$ and re-use the BFS-structure for the next iterative call of HungarianUpdate until we reach an unmatched worker, a player in R, or a firm with payoff zero in $\mathcal{O}(n^2)$. □

The complexity argument in the above proof is straight-forward. Further investigations might lead to slight improvements, though. Similar to the Hungarian Method one could keep the set $\bar{P} \cup \bar{Q}$ of vertices reachable from unmatched firms in a tree-structure until the matching is augmented. This tree changes when a rigid edge is matched. Then the "hungarian" tree rooted by an unmatched firm becomes a tree which is rooted by a matched player. We had no idea of a data structure that could help to efficiently recycle the data.

The advantage of the implementation discussed in the proof of Theorem 4.4 is that in any stage of the hungarian update one keeps track of the "distance" of unmatched and rigid players from the current component $\bar{P} \cup \bar{Q}$. Those values can be updated in linear time when the payoffs in $\bar{P} \cup \bar{Q}$ are altered by Δ. In the situation described in the above paragraph it seems to be difficult to update this data in linear time. Such a linear time update would lead to an $\mathcal{O}(n^3)$-algorithm having the same complexity as the Hungarian Method.

References

[1] Gabrielle Demange, David Gale, and Marilda Sotomayor. Multi-item auctions. *Journal of Political Economy*, 94(4):863–872, 1986.

[2] Xiaotie Deng and Christos H. Papadimitriou. On the complexity of cooparative game solution concepts. *Mathematics of Operations Research*, 19: 257–266, 1994.

[3] Xiaotie Deng, Toshihide Ibaraki, and Hiroshi Nagamochi. Combinatorial optimization games. In *Proceedings of the 8th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 720–729, New Orleans, LA, 1997.

[4] Kimmo Eriksson and Johan Karlander. Stable matching in a common generalization of the marriage and assignment models. *Discrete Mathematics*, 217(1-3):135–156, 2000.

[5] Ulrich Faigle, Sandor P. Fekete, Winfried Hochstättler, and Walter Kern. On the complexity of testing membership in the core of min cost spanning tree games. *International Journal of Game Theory*, 26:361–366, 1997.

[6] András Frank. On Kuhn's Hungarian method – A tribute from Hungary. Technical report, Egerváry Research Group on Combinatorial Optimization, October 2004.

[7] Saturo Fujishige and Akihisa 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.

[8] David Gale and Lloyd S. Shapley. College admissions and the stability of marriage. *American Mathematical Monthly*, 69:9–15, 1962.

[9] Zvi Galil. Efficient algorithms for finding maximum matching in graphs. *ACM Computing Surveys*, 18(1):23–38, 1986.

[10] Dan Gusfield and Robert W. Irving. *The stable marriage problem: Structure and algorithms.* MIT Press, Cambridge, MA, USA, 1989.

[11] Winfried Hochstättler and Robert Nickel. Note on an auction procedure for matching games in polynomial time. Technical report, FernUniversität in Hagen, Germany, 2005. In Preparation.

[12] Donald E. Knuth. Stable marriage and its relation to other combinatorial problems. In *CRM Proceedings and Lecture Notes*, volume 10. American Mathematical Society, 1997.

[13] Harold W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quaterly*, 2:83–97, 1955.

[14] Alvin E. Roth and Marilda Sotomayor. Stable outcomes in discrete and continuous models of two-sided matching: A unified treatment. *Revista de Econometria, The Brazilian Review of Econometrics*, 16(2), November 1996.

[15] Alwin E. Roth and Marilda Sotomayor. *Two-sided matching: A study in game-theoretic modeling and analysis.* Cambridge University Press, Cambridge, 1991.

[16] Lloyd S. Shapley and Martin Shubik. The assignment game I: The core. *International Journal of Game Theory*, 1:111–130, 1972.

[17] Marilda Sotomayor. Existence of stable outcomes and the lattice property for a unified matching market. *Mathematical Social Sciences*, 39:119–132, 2000.