Note on an Auction Procedure for a Matching Game in Polynomial Time

Winfried Hochstättler and Robert Nickel
Department of Mathematics
FernUniversität in Hagen
D-58084 Hagen

October 26, 2005

Abstract

We derive a polynomial time algorithm to compute a stable solution in a mixed matching market from an auction procedure as presented by Eriksson and Karlander [2]. As a special case we derive an $\mathcal{O}(nm)$ algorithm for bipartite matching that does not seem to have appeared in the literature yet.

1 Introduction

We consider a mixed matching market as introduced by Eriksson and Karlander [2] which generalizes Stable Marriage [5] and the Assignment Game [10].

This note was developed in parallel with [8] where we derive another polynomial time algorithm, to compute a stable solution for the same model, from the key lemma in Sotomayor [11]. In that paper a stable solution is shown to exist. Both algorithms run in $\mathcal{O}(n^4)$, where $2n$ is the number of players and n^2 is the size of a problem instance.

In the next section we briefly introduce the model and its notion of stability. Then we design a polynomial time algorithm to compute a stable solution in Section 3. Finally, we discuss the behaviour of the algorithm in the special cases of Stable Matching, Assignment Game and cardinality matching and summarize differences from and similarities to the algorithm from [8].

2 Notation

We have two sets of players P (firms indexed by i) and Q (workers indexed by j) w.l.o.g. satisfying $|P| = |Q| =: n$. Let furthermore $P \cup Q$ be partitioned into *flexible* players (F) and *rigid* players (R). Consider the complete bipartite graph on $P \dot{\cup} Q$. An edge (i, j) is called *rigid* if one of i or j is in R and *flexible*, otherwise. For each edge (i, j) there are nonnegative numbers a_{ij} and b_{ij}. The sum $a_{ij} + b_{ij}$ is the *productivity* of a cooperation between i and j. If i cooperates with j and (i, j) is a free edge and this productivity can be freely divided into *payoffs* u_i and v_j while $u_i = a_{ij}$ and $v_j = b_{ij}$ must hold if (i, j) is a rigid edge.

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

(i) $u_i + v_j \geq a_{ij} + b_{ij}$ *if (i, j) is a free edge and*

(ii) $u_i \geq a_{ij}$ *or* $v_j \geq b_{ij}$ *if (i, j) is a rigid edge.*

A stable outcome *is a stable payoff (u, v) together with a bijective map $\mu : P \to Q$ (denoted by $(u, v; \mu)$) so that*

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

(iv) $u_i + v_j = a_{ij} + b_{ij}$ *for $\mu(i) = j$ and $\{i, j\} \subseteq F$.*

(v) $u_i = a_{ij}$ *and* $v_j = b_{ij}$ *for $\mu(i) = j$ and $\{i, j\} \cap R \neq \emptyset$.*

Let $\mu : P \to Q$ be a map. If $\mu(i) = j$ then we say *i proposes to j*. A proposal is called *free* or *rigid* if the corresponding edge is free resp. rigid. Any firm $i \in \mu^{-1}(Q)$ is called *mapped.* If there are firms i_1, i_2 so that $\mu(i_1) = \mu(i_2) = j$ then j is called *doubly mapped.* We denote by

Q_U the set of unmapped workers,
$Q_{2\mu}$ the set of doubly mapped workers,
Q_R the set of workers that have a rigid proposal, and by
Q_{2R} the set of workers with at least 2 rigid proposals.

Let furthermore

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

define the possible profit of i from j if j receives v_j.

The strategy of the algorithm is the following: The map μ always defines stable relations but is not necessarily injective. In the course of the algorithm we will try and increase $\mu(P)$, keeping stability of the relations, until the map is injective. The procedure to increase $\mu(P)$ acts on the *augmentation digraph* $G^{(v,\mu)} = (P \cup Q, A)$ with backward arcs (j, i) for $\mu(i) = j$ and forward arcs (i, j) for $j \in D_i^{(v,\mu)}$ where

$$D_i^{(v,\mu)} = \{j \in Q \mid f_{ij} = \max_k f_{ik}\}$$

is the set of workers that maximize the potential benefit of firm i. 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_s is called *alternating* resp. *augmenting* if j_s is not mapped.

3 An Algorithm to Find a Stable Outcome

Eriksson and Karlander [2] assume integer data and in one step increase a free payoff by at most one. We modify this approach in such a way that we increase the payoff by the smallest possible amount that changes the augmentation digraph. Our strategy to make the map $\mu : P \to Q$ bijective is as follows: As in the classical "men-propose-women-dispose" algorithm from Gale and Shapley [5] workers with more than one rigid proposal choose the best one and dispose the rest. This way some firms become temporarily unmapped. Each of these unmapped firms has to place another proposal until every worker has at most one rigid proposal. Next, we search the graph $G^{(v,\mu)}$ for alternating paths that reach a worker in $Q_U \cup Q_R$ and alternate the matching. If none of the above is possible, we increase the payoffs v of workers which are reachable by an alternating path until $G^{(v,\mu)}$ receives a new edge and the process is repeated until the map becomes injective.

The algorithm uses several sub-procedures:

Propose(i): Places a proposal from i to a worker in $D_i^{(v,\mu)}$, i. e. chooses $\mu(i) \in D_i^{(v,\mu)}$.

Dispose(j, i^*): Disposes all firms $i \neq i^*$ that made a rigid proposal to j, i. e. sets $\mu(i)$ to be undefined for all $i \in \mu^{-1}(j) \setminus \{i^*\}$.

Alternate($\mathcal{P}$): μ is alternated along the alternating path $\mathcal{P}$, i. e. all arcs are reoriented and μ is modified such that it uses the new backward arcs. If $\mathcal{P}$ is augmenting then the size of the image of μ increases by 1.

BFS(G, Q_{2R}): Returns all vertices reachable from Q_{2R} in G.

PlaceRigidProposals: This procedure is the "men propose – women dispose" algorithm of Gale and Shapley [5]. Here, we denote by P_U the set of temporarily unmapped firms. See Algorithm 2.

HungarianUpdate: Increases the payoffs of all workers reachable from a doubly mapped worker. See Algorithm 3 for details.

Algorithm 1 An Algorithm to Find a Stable Outcome

```
$v \leftarrow 0$
PlaceRigidProposals
while $Q_{2\mu} \neq \emptyset$ do
    while $\exists$ augmenting path to $j \in (Q \setminus \mu(P)) \cup Q_R$ do
        Alternate($\mathcal{P}$)
        PlaceRigidProposals
    end while
    HungarianUpdate
end while
```

Algorithm 2 PlaceRigidProposals

```
while $P_U \neq \emptyset$ do
    for all $i \in P_U$ do
        Propose($i$)
    end for
    for all $j \in Q_{2R}$ do
        Let $i^*$ be the favorite proposal in $\mu^{-1}(j)$
        Dispose($j, i^*$)
        $v_j \leftarrow b_{i^* j}$
    end for
end while
```

Algorithm 3 HungarianUpdate

```
$\bar{P}\dot{\cup}\bar{Q} \leftarrow$BFS($G^{(v,\mu)}, Q_{2R}$)
$u_i \leftarrow \max_j f_{ij}$
$\Delta \leftarrow \min\{u_i - f_{ik} \mid i \in \bar{P},\, k \in Q \setminus \bar{Q}\}$
for all $j \in \bar{Q}$ do
    $v_j \leftarrow v_j + \Delta$
end for
```

Theorem 3.1. *Algorithm 1 eventually finishes with a stable outcome and can be implemented to run in $\mathcal{O}(n^4)$ time.*

Proof. In any iteration of the inner loop of line 4 in Algorithm 1 $|\mu(P)|$ is increased or a rigid proposal is disposed. If there is a path to $Q \setminus \mu(P)$ then $|\mu(P)|$ increases. If the path ends in $j \in Q_R$ then PlaceRigidProposals is called and disposes at least one rigid edge. Note, that a rigid edge once disposed will never be proposed again. If no path exists at all then v is increased by HungarianUpdate until this is the case and in each call of HungarianUpdate at least one new arc shows up in $G^{(v,\mu)}$. Thus, the procedure is finite.

PlaceRigidProposals can be implemented to run in $\mathcal{O}(n^2)$ (see e. g. Gusfield and Irving [7]). The while-loop in line 4 of Algorithm 1 might be iterated more than once without finding a path as desired. Anyway, HungarianUpdate can be implemented so that its consecutive calls until a path is found need $\mathcal{O}(n^2)$ time in sum by reusing the BFS-structure from the previous call and storing a minimum distance Δ_j from unmapped vertices and vertices in Q_R to the current BFS forest (see e. g. Galil [6] or Hochstättler et al. [8] for details). Hence, after $\mathcal{O}(n^2)$ time steps we can augment μ or dispose a rigid edge which can happen at most $\mathcal{O}(n^2)$ times.

Next we will show that the algorithm produces a stable outcome. In any stage of the algorithm let $\bar{u}_i := \max_j f_{ij}^{(v,\mu)}$. Then $(\bar{u}, v)$ is stable and $(\bar{u}, v; \mu)$ satisfies (iv) and (v) of Definition 2.1 since $\mu(i) = j$ implies $j \in D_i^{(v,\mu)}$. As v monotonically increases we also have $v \geq 0$. A worker with no proposer always has payoff zero and is therefore of non-negative value to all firms. Hence together with (iv) and (v) this implies $u \geq 0$. When the algorithm terminates μ is bijective and thus, $(\bar{u}, v; \mu)$ is a stable outcome. □

4 Special Cases and Remarks

When $F = \emptyset$ our model coincides with the Stable Marriage Model, since the a_{ij} at firm i resp. b_{ij} at worker j may be replaced by preference lists. Since $Q_{2R} = \emptyset$ implies $Q_{2\mu} = \emptyset$ the algorithm stops after the execution of PlaceRigidProposals. Inside the while-loop of Algorithm 2 mapping edges to a worker in Q_{2R} are disposed and by updating v_j never become proposals again. Here, proposals are made in rounds. In any round a firm proposes to its favorite worker. Then every worker choses his best proposal and disposes the other firms. This corresponds to the famous "men propose – women dispose" introduced by Gale and Shapley [5].

If $R = \emptyset$ and $a_{ij} + b_{ij} \in \{0, 1\}$ for any edge (i, j) the problem reduces to finding a matching of maximum cardinality among edges with productivity 1 (refered to as *1-edges*). The presented algorithm does not seem to have appeared in the literature yet and differs from the standard approach which starts with an empty matching and searches the graph of 1-edges for an augmenting path. The algorithm presented here starts with a total but not surjective (and therefore not injective) map μ. An alternating path from a doubly mapped worker to an unmapped worker increases the size of the image of μ until μ is bijective, i. e. a matching. While the standard approach is essentially due to Ford and Fulkerson [3] the approach presented here reminds of the preflow-push algorithm (see e. g. [1]). However, the strategy of lifting node potentials in preflow-push does not seem to have anything in common with the augmenting path procedure used here. A naive implementation would lead to an $\mathcal{O}(nm)$ algorithm. We wonder if this approach might lead to efficient implementations for cardinality matching competing with existing algorithms.

If $R = \emptyset$ the algorithm reminds of the Hungarian Method. Like the latter our method is a primal-dual algorithm but the subroutine for cardinality matching is replaced as described above. The algorithm starts with a weighted vertex cover (u, v) if we set $u_i \leftarrow \max_j f_{ij}$. The Hungarian Method in all its variants searches for a path augmenting a given matching or increases the values v_j until such a path can be found (see Frank [4] for a transparent presentation). The search is performed on a digraph where the matching edges are the backward arcs. The main difference between our algorithm (for $R = \emptyset$) and the Hungarian Method is the graph on which we look for alternating resp. augmenting paths as we have all the mapping edges as backward arcs. Thus, our search tree in average should be shorter.

The algorithm in [8] to find a stable outcome differs from the algorithm presented here in various ways. In [8] (especially rigid) proposals are made asynchronously. Furthermore, this algorithm is a direct extension of the Hungarian Method as introduced in Kuhn [9, Variant 2], while the algorithm presented here is much closer to the original "men propose – women dispose" algorithm of Gale and Shapley [5]. Also the concepts of augmenting paths differ as here we augment maps and there matchings are augmented. Nevertheless, the directed search graphs have the same underlying graph.

References

[1] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows.* Prentice Hall, 1993.

[2] 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.

[3] Lester R. Ford and Delbert R. Fulkerson. A simple algorithm for finding maximal network flows and an application to the hitchcock problem. *Canadian Journal of Mathematics*, 9:210–218, 1957.

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

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

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

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

[8] Winfried Hochstättler, Hui Jin, and Robert Nickel. The hungarian method in a mixed matching market. Technical report, FernUniversität in Hagen, Germany, October 2005.

[9] Harold W. Kuhn. Variants of the Hungarian method for the assignment problem. *Naval Research Logistics Quaterly*, 3:253–258, 1956.

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

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