# Maximum Cardinality Popular Matchings in Strict Two-sided Preference Lists

Telikepalli Kavitha and Chien-Chung Huang

December 4, 2010

MPI–I–2010–1-001 December 2010

## Authors' Addresses

Telikepalli Kavitha
Tata Institute of Fundamental Research
Homi Bhabha Road
Mumbai 400 005
India

Chien-Chung Huang
Max-Planck-Institut für Informatik
Campus E 1 4
D-66123 Saarbrücken
Germany

## Publication Notes

A part of this work was done when C.-C. Huang visited TIFR Mumbai under the IMPECS program.

## Acknowledgements

We thank Vincenzo Bonifaci for his comments on this work.

## Abstract

We consider the problem of computing a maximum cardinality *popular* matching in a bipartite graph $G = (\mathcal{A} \cup \mathcal{B}, E)$ where each vertex $u \in \mathcal{A} \cup \mathcal{B}$ ranks its neighbors in a strict order of preference. This is the same as an instance of the *stable marriage* problem with incomplete lists. A matching $M^*$ is said to be popular if there is no matching $M$ such that more vertices are better off in $M$ than in $M^*$.

Popular matchings have been extensively studied in the case of one-sided preference lists, i.e., only vertices of $\mathcal{A}$ have preferences over their neighbors while vertices in $\mathcal{B}$ have no preferences; polynomial time algorithms have been shown here to determine if a given instance admits a popular matching or not and if so, to compute one with maximum cardinality. It has very recently been shown that for two-sided preference lists, the problem of determining if a given instance admits a popular matching or not is NP-complete. However this hardness result assumes that preference lists have *ties*. When preference lists are *strict*, it is easy to show that popular matchings always exist since stable matchings always exist and they are popular. But the complexity of computing a maximum cardinality popular matching was unknown. In this paper we show an $O(mn)$ algorithm for this problem, where $n = |\mathcal{A}| + |\mathcal{B}|$ and $m = |E|$.

## Keywords

Matching under Preferences, Popular Matchings

## Contents

# 1 Introduction

Our input is a bipartite graph $G = (\mathcal{A} \cup \mathcal{B}, E)$ where each vertex ranks its neighbors in a strict order of preference. Each vertex $u \in \mathcal{A} \cup \mathcal{B}$ seeks to be assigned to one of its neighbors and $u$'s preference is given by the ordering in its preference list. Preferences can be incomplete, which means that each vertex may rank only a subset of vertices on the other side. (We assume without loss of genereaility that $a$ belongs to $b$'s list if and only if $b$ belongs to $a$' list.) Note that this is the same as an instance of the *stable marriage* problem with incomplete lists and it is customary to call the two sides of the graph *men* and *women* respectively. We use $V$ to denote the entire vertex set $\mathcal{A} \cup \mathcal{B}$ and $n$ for the number of vertices and $m$ for the number of edges in $G$. We assume that no vertex is isolated, so $m \geq n/2$.

A matching $M$ is a set of edges no two of which share an endpoint. An edge $(u, v)$ is said to be a *blocking edge* for a matching $M$ if by being matched to each other, both $u$ and $v$ are *better off* than their respective assignments in $M$: that is, $u$ is either unmatched in $M$ or prefers $v$ to $M(u)$ and similarly, $v$ is either unmatched in $M$ or prefers $u$ to $M(v)$. A matching that admits no blocking edges is called a stable matching. It is known that every instance $G$ admits a stable matching [9] and such a matching can be computed in linear time by a straightforward generalization [5] of the Gale-Shapley algorithm [3] for complete lists.

## 1.1 Popular Matchings

For any two matchings $M$ and $M'$, we say that vertex $u$ prefers $M$ to $M'$ if $u$ is better off in $M$ than in $M'$ (i.e., $u$ is either matched in $M$ and unmatched in $M'$ or prefers $M(u)$ to $M'(u)$). We say that $M$ is more popular than $M'$, denoted by $M \succ M'$, if the number of vertices that prefer $M$ to $M'$ is more than the number of vertices that prefer $M'$ to $M$.

**Definition 1.** *A matching $M$ is popular if there is no matching that is more popular than $M$.*

Popularity is an attractive notion of optimality as a majority vote cannot force a migration from a popular matching. Gardenfors introduced the notion of popularity in the context of stable matchings.

Popular matchings have been studied extensively during the last few years in the case where only vertices in $\mathcal{A}$ have preferences while vertices in $\mathcal{B}$ have no preferences. Thus each edge $e = (a, b)$ in $G$ has a rank associated with it (the rank that $a$ assigns to $b$). There are simple examples of instances in the one-sided preference lists domain that admit no popular matching; Abraham et al. [1] gave efficient algorithms to determine if a given instance admits a popular matching or not and if so, to compute one with maximum cardinality. In fact, for one-sided preference lists (both for strict lists and for lists with ties), they gave a structural characterisation of instances that admit popular matchings.

In the case of two-sided preference lists with ties, it has recently been shown by Biró, Irving, and Manlove [2] that the problem of deciding if an instance admits a popular matching or not is NP-complete. It is important to have ties in preference lists for the NP-hardness result because in the absence of ties (i.e., when all the preference lists are strict), a stable matching is popular.

While comparing a stable matching $S$ to any matching $M$, note that for any edge $e \in M$, *both* the endpoints of $e$ cannot prefer $M$ to $S$ - if they do, then it contradicts the stability of $S$. Hence if one endpoint of $e$ prefers $M$ to $S$, then the other has to prefer $S$ to $M$. Thus the number of votes in favor of $M$ is at most the number of votes in favor of $S$, hence $M$ cannot be more popular than $S$. Thus popular matchings always exist in the world of two-sided strict preference lists. But not all popular matchings are stable as shown by this simple example: let $\mathcal{A} = \{a_1, a_2\}$ and $\mathcal{B} = \{b_1, b_2\}$ and let the preference lists be as shown below:

$$a_1 : b_1, b_2, \qquad a_2 : b_1, \qquad b_1 : a_1, a_2, \qquad \text{and} \qquad b_2 : a_1.$$

Here $a_1$'s top choice is $b_1$ and second choice is $b_2$ while $a_2$ has a single neighbor $b_1$. The vertex $b_1$'s top choice is $a_1$ and second choice is $a_2$ while $b_2$ has a single neighbor $a_1$. In this instance, the matching $\{(a_1, b_1)\}$ is the only stable matching, while the matching $\{(a_1, b_2), (a_2, b_1)\}$ is popular but unstable.

## Our problem.

Stability is an important and well-accepted notion of optimality while computing a matching in $G = (\mathcal{A} \cup \mathcal{B}, E)$ with 2-sided strict preference lists. Since $\{$popular matchings$\} \supseteq \{$stable matchings$\}$, stability is a stronger concept. But there are many problems, where blocking edges may be permitted as long as there is no *majority* of vertices who are better-off in another matching, for instance in allocating training positions to trainees or projects to students. In such problems popularity becomes a natural optimality criterion.

Also, popularity scores over stability with respect to the size of the maximum matching possible under this optimality criterion. As seen in the above example, the given instance could have popular matchings whose size is strictly larger than the size of a stable matching. It is known that all stable matchings in $G = (\mathcal{A} \cup \mathcal{B}, E)$ have the same size and match exactly the same set of vertices ([5], Section 4.5.2). Thus in problems that seek to match as many vertices as possible under this weaker notion of stability that $M$ is acceptable if there is *no matching where more vertices are better off* compared to $M$, what we seek is a maximum cardinality popular matching. We consider this problem in this paper and show the following result.

**Theorem 1.** *A maximum cardinality popular matching in a bipartite graph $G = (\mathcal{A} \cup \mathcal{B}, E)$ with 2-sided strict preference lists can be computed in $O(mn)$ time, where $m$ is the number of edges and $n$ is the number of vertices in $G$.*

In order to construct such a matching in $G$, we show a sufficient condition for a matching to be a maximum cardinality popular matching. We need the following definition first:

**Definition 2.** *For any vertex $u \in \mathcal{A} \cup \mathcal{B}$ and neighbors $x$ and $y$ of $u$, define $u$'s vote between $x$ and $y$ as follows:*

$$\mathsf{vote}_u(x,y) = \begin{cases} 1 & \text{if } u \text{ prefers } x \text{ to } y \\ -1 & \text{if } u \text{ prefers } y \text{ to } x \\ 0 & \text{otherwise (i.e., } x = y\text{).} \end{cases}$$

**Our approach.** Suppose we partition the vertex set $\mathcal{A} \cup \mathcal{B}$ into $L$ and $R = V \setminus L$ and reorganize the graph $G$ by placing all the vertices of $L$ on the left and all the vertices of $R$ on the right. Note that $L$ and $R$ need not be independent sets. Thus we could have edges between vertices of $L$ and similarly, edges between vertices of $R$. Let $M$ be a matching in $L \times R$, i.e., every edge of $M$ has one endpoint in $L$ and the other endpoint in $R$.

**Definition 3.** *Call a matching $M \subseteq L \times R$* good *with respect to $(L,R)$ if the following two properties are satisfied:*

*(1) There is no edge $(a,b) \in L \times R$ such that $\mathsf{vote}_a(b, M(a)) = 1$ and $\mathsf{vote}_b(a, M(b)) = 1$.*

*(2) If $(a,b) \in L \times L$, then $\mathsf{vote}_a(b, M(a)) = -1$ and $\mathsf{vote}_b(a, M(b)) = -1$.*

Note that in case $u$ is unmatched in $M$, then $\mathsf{vote}_u(v, M(u)) = 1$ for any neighbor $v$ of $u$, since every vertex prefers being matched to being unmatched.

Property (1) of goodness states that there is no *unstable* edge in $L \times R$ for $M$. Property (2) of goodness states that for every $e$ in $L \times L$, each endpoint of $e$ prefers its partner in $M$ to the other endpoint of $e$. Our main theorem is the following.

**Theorem 2.** *Let $M$ be a matching that is good with respect to some partition $(L,R)$ of $V$. If every vertex of $R$ is matched in $M$, then $M$ is a maximum cardinality popular matching.*

We build such a matching $M$ in our algorithm. The vertices in $R$ can be viewed as the "sought-after" vertices since they are all matched in $M$ and the vertices in $L$ are the vertices that *seek* partners in $R$. Our algorithm is iterative: each iteration involves 2 invocations of a Gale-Shapley type of proposal-disposal algorithm between the current $L$ and $R$. Throughout the algorithm, we maintain the invariant that our matching is good with respect to the current partition $(L,R)$. We show that termination (every vertex in the current $R$ getting matched) has to happen within the first $n$ iterations, thus the running time of our algorithm will be $O(mn)$.

## 1.2 Background

Subsequent to the work in [1] on one-sided popular matchings, there have been several variants of the popular matchings problem considered in the domain of one-sided preference lists. One line of research has been on generalizations of the popular matchings problem while the other direction has been to deal with instances that do not admit any popular matchings. The generalizations include the capacitated version studied by Manlove and Sng [11], the weighted version studied by Mestre [14] and random popular matchings considered by Mahdian [10]. Kavitha and Nasre [8] as well as McDermind and Irving [13] independently studied the problem of computing an optimal popular matching for strict instances where the notion of optimality is specified as a part of the input. For instances that do not admit popular matchings, McCutchen [12] considered the problem of computing a least unpopular matching and showed this problem to be NP-hard while Kavitha, Mestre, and Nasre [7] showed the existence of popular mixed matchings and efficient algorithms for computing them.

The problem of computing popular matchings and its variants in the domain of two-sided preference lists has not received much attention so far. To the best of our knowledge, the only works dealing with popular matchings in instances with two-sided preference lists are by Gardenfors [4], who originated the notion of popular matchings, and by Biró et al. [2] who showed NP-hardness results for the problems of computing an arbitrary popular matching and a maximum cardinality popular matching for preference lists with ties. It was shown in [2] that the problem of determining if a given bipartite graph $G$ admits a maximal matching of size $k \in \mathbb{Z}$ can be reduced to the problem of deciding if an augmented version of $G$ with ties in preference lists admits a popular matching. As the former problem is NP-hard, so is the latter problem. It was not known so far if the maximum cardinality popular matching problem admits a polynomial time algorithm.

**Organization of the paper.** Section 2 contains the proof of Theorem 2 and Section 3 has our algorithm and its proof of correctness. We conclude in Section 4.

# 2 Our sufficient condition

Section 2.1 contains the proof of Theorem 2 and Section 2.2 briefly discusses our method to construct a *good* matching that satisfies the condition given in Theorem 2.

## 2.1 Proof of Theorem 2

We need to show that if $M$ is a matching that is good with respect to some partition $(L,R)$ of $V$ and $M$ matches all the vertices of $R$, then $M$ is a maximum cardinality popular matching. Let $M'$ be any matching in $G$. Define $\Delta(M',M)$ as follows:

$$\Delta(M',M) = \sum_{u\in\mathcal{A}\cup\mathcal{B}} \mathsf{vote}_u(M'(u),M(u)).$$

Thus $\Delta(M',M)$ is the difference between the votes that $M'$ gets and the votes that $M$ gets. Note that $M(u)$ or $M'(u)$ can be also the state of being unmatched, which is the least preferred state for any $u$.

We will show that $\Delta(M',M) \le 0$ for any matching $M'$ and in particular, if $|M'| > |M|$, then $\Delta(M',M) < 0$. This implies that $M$ is popular and every matching whose size is more than $|M|$ is unpopular. In other words, there is no popular matching in $G$ whose size is more than $|M|$.

Let $\rho$ be any connected component in $M \oplus M'$. Since $\Delta(M',M) = \sum_\rho \sum_{u\in\rho} \mathsf{vote}_u(M'(u),M(u))$, we will compute $\sum_{u\in\rho} \mathsf{vote}_u(M'(u),M(u))$ for any $\rho \in M \oplus M'$ now. Note that each $\rho \in M \oplus M'$ is either a cycle or a path.

**Lemma 3.** *If* $\rho$ *is a cycle, then* $\sum_{u\in\rho} \mathsf{vote}_u(M'(u),M(u)) \le 0$.

*Proof.* Let us pair the vertices of $\rho$ along the edges of $M' \cap \rho$. We partition the edges of $M' \cap \rho$ into three categories: $L \times R$ edges, $R \times R$ edges, and $L \times L$ edges. First, for every edge $e$ of $M' \cap \rho$ that is in $L \times R$, the sum of votes of the endpoints of $e$ for $M'$ vs $M$ is at most 0 since $M$ is good (Definition 3, property (1)).

Next, since $M$ uses only edges of $L \times R$ and because every vertex in $\rho$ is matched in $M$, the number of $L$ vertices in $\rho$ equals the number of $R$ vertices in $\rho$. Hence if $M' \cap \rho$ has $k$ edges of $R \times R$ (call these $e_1,\ldots,e_k$), then $M' \cap \rho$ also has $k$ edges of $L \times L$ (call these $e'_1,\ldots,e'_k$).

By property (2) of the goodness of $M$, every endpoint $u$ of an $L \times L$ edge $(u,v)$ votes $-1$ for $v = M'(u)$ vs $M(u)$. Hence the sum of votes of the endpoints of $e'_1,\ldots,e'_k$ (for $M'$ vs $M$)

is $-2k$. Thus even if every endpoint of the edges $e_1,\dots,e_k$ votes $+1$ for $M'$ (vs $M$), the total sum of votes of the endpoints of $e_1,\dots,e_k$ and the endpoints of $e'_1,\dots,e'_k$ is at most 0. Thus $\sum_{u\in\rho}\mathsf{vote}_u(M'(u),M(u)) \le 0$. □

**Lemma 4.** *If* $\rho$ *is a path, then* $\sum_{u\in\rho}\mathsf{vote}_u(M'(u),M(u)) \le 0$.

*Proof.* As done in Lemma 3, here also we will sum the votes of the vertices in $\rho$ by grouping them as the endpoints of $L\times R$ edges, the endpoints of $L\times L$ edges, and the endpoints of $R\times R$ edges. Depending on whether $|\rho\cap M'|$ is larger than/equal to/smaller than $|\rho\cap M|$, we have the following three claims:

**Claim 1.** *If* $|\rho\cap M'| > |\rho\cap M|$, *then* $\sum_{u\in\rho}\mathsf{vote}_u(M'(u),M(u)) < 0$.

Every vertex of the path $\rho$ is matched in $M'$. Thus $\rho$ is an augmenting path with respect to $M$. Since $M$ leaves no vertex of $R$ unmatched, both the endpoints of $\rho$ have to be $L$ vertices; so $\rho$ has $t$ vertices of $R$ and $t+2$ vertices of $L$, where $t=|\rho\cap M|$.

Thus if the number of $R\times R$ edges in $\rho\cap M'$ is $k$, then the number of $L\times L$ edges in $\rho\cap M'$ is $k+1$ because every vertex in $\rho$ is matched by $M'$. Hence $\sum_{u\in\rho}\mathsf{vote}_u(M'(u),M(u)) < 0$ by partitioning this sum into the endpoints of $L\times R$ edges (their sum of votes is at most 0), the endpoints of $L\times L$ edges (their sum of votes is $-2k-2$), and the endpoints of $R\times R$ edges (their sum of votes is at most $2k$).

**Claim 2.** *If* $|\rho\cap M'| = |\rho\cap M|$, *then* $\sum_{u\in\rho}\mathsf{vote}_u(M'(u),M(u)) < 0$.

Let $t=|\rho\cap M|$. So $\rho$ has length $2t$. Thus $\rho$ has $2t+1$ vertices, of which one is left unmatched in $M$ and the remaining $2t$ are matched in $M$. Since it is given that $M$ matches all the vertices of $R$, the path $\rho$ has $t$ vertices of $R$ and $t+1$ vertices of $L$ (the $2t$ vertices of the $L\times R$ edges of $M$ and the one unmatched vertex of $L$).

So the number of $L$-vertices of $\rho$ that are matched in $M'$ is at least $t$ and the number of $R$-vertices of $\rho$ matched in $M'$ is at most $t$. Hence if the number of $R\times R$ edges in $\rho\cap M'$ is $k$, then the number of $L\times L$ edges in $\rho\cap M'$ is at least $k$. It is now immediate that $\sum_{u\in\rho}\mathsf{vote}_u(M'(u),M(u)) \le 0$, by partitioning $\sum_{u\in\rho}\mathsf{vote}_u(M'(u),M(u))$ into the endpoints of $L\times R$ edges, the endpoints of $L\times L$ edges, and the endpoints of $R\times R$ edges.

In fact, $\sum_{u\in\rho}\mathsf{vote}_u(M'(u),M(u)) < 0$ since there is also a vertex $v$ in $\rho$ that is unmatched in $M'$ and $v$ votes $-1$ for $M'$ (vs $M$) since $v$ is matched in $M$.

**Claim 3.** *If* $|\rho\cap M'| < |\rho\cap M|$, *then* $\sum_{u\in\rho}\mathsf{vote}_u(M'(u),M(u)) \le 0$.

Every vertex of $\rho$ is matched in $M$ and thus $\rho$ has $t$ vertices of $R$ and $t$ vertices of $L$, where $t=|\rho\cap M|$. Thus if the number of $R\times R$ edges in $\rho\cap M'$ is $k$, then the number of $L\times L$ edges in $\rho\cap M'$ is at least $k-1$.

If the number of $L\times L$ edges in $\rho\cap M'$ is $k$ or more, then the same argument as in the proofs of Claims 1 and 2 holds here too and thus $\sum_{u\in\rho}\mathsf{vote}_u(M'(u),M(u)) \le 0$. In case the number of $L\times L$ edges in $\rho\cap M'$ is $k-1$, then the sum of votes (for $M'$ vs $M$) of the vertices of $\rho$ matched in $M'$ is at most 2; also, there are 2 vertices of $L$ that are matched in $M$ and are left unmatched in $M'$ and they contribute 2 votes of $-1$ each for $M'$ vs $M$. So $\sum_{u\in\rho}\mathsf{vote}_u(M'(u),M(u)) \le 0$.

This finishes the proof of Lemma 4. □

Since Lemmas 3 and 4 hold for every $\rho$ in $M \oplus M'$, it follows that $\Delta(M', M) \leq 0$. Thus $M$ is popular. We have also shown that if $|M'| > |M|$, then $\Delta(M', M) < 0$ (by Claim 1), that is, $M$ is more popular than $M'$. In other words, any matching of size more than $M$ is unpopular, thus $M$ is a maximum cardinality popular matching. This finishes the proof of Theorem 2.

The following is an interesting corollary that follows from the proof of Theorem 2.

**Corollary 5.** *Suppose there exists a matching $M$ that is good with respect to some partition $(L, R)$ and $M$ matches all the vertices of $R$. Then every maximum cardinality popular matching in $G$ matches the same vertices as $M$.*

*Proof.* Suppose $M' \oplus M$ contains a path, call it $\rho$.

- If $|\rho \cap M'| \geq |\rho \cap M|$, then $M'$ is unpopular (by Claims 1 and 2).
- If $|\rho \cap M'| < |\rho \cap M|$, then $|M'| < |M|$, so $M'$ cannot be a maximum cardinality popular matching.

Hence if $M'$ is also a maximum cardinality popular matching, then $M' \oplus M$ has to be a set of cycles, that is, $M'$ matches the same vertices as $M$. □

## 2.2 Implementing our sufficient condition

The main question to be answered now is: does there always exist a matching that is good with respect to some $(L, R)$ that matches all the vertices in $R$? We answer this question positively in the next section by constructing such a matching. In this section we give some intuition.

Let $S$ be a stable matching in $G = (\mathcal{A} \cup \mathcal{B}, E)$. We assume that $S$ does not match all the vertices of $\mathcal{A} \cup \mathcal{B}$. (If it does, then $S$ is the matching that we seek.)

Let $A_0 \subset \mathcal{A}$ and $B_0 \subset \mathcal{B}$ be the set of men and women, respectively, that are unmatched in $S$. Let $L_1 = A_0 \cup B_0$ and let $R_1$ be $V \setminus L_1$ (thus $R_1$ is the set of vertices that are matched in $S$). Let us first construct a matching $M_1$ that is *good* with respect to the partition $(L_1, R_1)$. How do we ensure property (1) of the definition of goodness? The answer lies in computing $M_1$ as a matching where vertices of $L_1$ propose to the vertices of $R_1$ and vertices of $R_1$ dispose. That is, we run Gale-Shapley algorithm on the "bipartite" graph obtained by placing $L_1$ on the left and $R_1$ on the right and the edge set restricted to $E \cap (L_1 \times R_1)$. For the sake of completeness, we present the Gale-Shapley proposal-disposal algorithm between any $L \subset \mathcal{A} \cup \mathcal{B}$ and $R = V \setminus L$ below in Figure 2.1.

Since every $\ell \in L$ proposes in decreasing order of preference and every $r \in R$ improves in the choice of its partner whenever $M(r)$ gets reassigned, the following claim is straightforward.

**Claim 4.** *If $M$ is the matching returned by the above algorithm, then there is no edge $(\ell, r)$ in $L \times R$ such that $\mathsf{vote}_\ell(r, M(\ell)) = 1$ and $\mathsf{vote}_r(\ell, M(r)) = 1$.*

Hence if $M_1$ is the matching obtained by the proposal-disposal algorithm between $L_1$ and $R_1$, then $M_1$ is *good* with respect to the partition $(L_1, R_1)$. This is because property (1) of goodness holds by the very nature of the proposal-disposal algorithm and property (2) of the

```
– $M = \emptyset$.
while there is some $u \in L$ unmatched in $M$ who has not yet been rejected by all its neighbors in $R$ do
    – $u$ proposes to its most preferred neighbor $v \in R$ that has not rejected $u$.
    if $v$ prefers $u$ to $M(v)$ then
        – $v$ assigns $M(v) = u$.    {so the vertex that was $v$'s previous partner in $M$ is now rejected by $v$}
    else
        – $v$ rejects $u$.
    end if
end while
– Return $M$.
```

Figure 2.1: Computing a matching $M \subseteq L \times R$ with $L$ proposing and $R = V \setminus L$ disposing

goodness of any matching $M_1 \subseteq L_1 \times R_1$ is vacuously true, since $L_1$ is an independent set. Recall that $L_1 = A_0 \cup B_0$ is the set of vertices left unmatched in the stable matching $S$, hence there is no edge with both endpoints in $L_1$.

Thus if $M_1$ matches all the vertices of $R_1$, then we have our desired matching. Otherwise, we enter the *second stage* of the first iteration. In the second stage, we move all the unmatched *men* from $R_1$ to $L_1$ and run the proposal-disposal algorithm between the new $L_1$ and the new $R_1$ to compute $M_1'$. It can be shown that $M_1'$ is good with respect to the new left-right partition.

If $M_1'$ matches all the vertices on the right, then this is the desired matching. Otherwise let $B_1$ denote the set of unmatched vertices on the right who are not matched by $M_1'$. We set $L_2 = L_1 \cup B_1$ and $R_2 = R_1 \setminus B_1$ and move to the next iteration of the algorithm.

Note that the men who moved from right to left at the end of the first stage are back on the right now in $R_2$. It is important to move them back to $R_2$ to ensure that the new matching $M_2$ that will be obtained in the first stage of the next iteration will be *good* with respect to $(L_2, R_2)$. Nevertheless, it was important to first move them to the left to identify $B_1$. The fact that $B_1$ was unmatched on the right in spite of the presence of these men on the left will be crucial in maintaining the invariant that the future matchings that we compute during the course of the algorithm will be good. As soon as we have a matching that matches all the vertices on the right, the algorithm terminates.

# 3 The Algorithm

Our algorithm to find a matching $M$ that fulfils the conditions given in Theorem 2 is given in Fig. 3.1. Recall that these conditions are:

- $M$ is good with respect to some partition $(L,R)$ and
- $M$ matches all the vertices of $R$.

**Algorithm 1** *Input:* $G = (\mathcal{A} \cup \mathcal{B}, E)$ *with strict preference lists*

```
1.  Let $S$ be the stable matching obtained by the proposal-disposal algorithm
    on $(\mathcal{A}, \mathcal{B})$. $\{$That is, men propose and women dispose.$\}$
2.  Set $L_1 =$ vertices left unmatched in $S$ and $R_1 = V \setminus L_1$.
3.  $i = 1$.
4.  while true do
5.      compute a matching $M_i$ by the proposal-disposal algorithm on $(L_i, R_i)$.
6.      if $M_i$ matches all of $R_i$ then return $M_i$.
7.      let $A_i \subset \mathcal{A}$ be the men in $R_i$ who are unmatched in $M_i$.
8.      set $L'_i = L_i \cup A_i$ and $R'_i = V \setminus L'_i$.
9.      compute a matching $M'_i$ by the proposal-disposal algorithm on $(L'_i, R'_i)$.
10.     if $M'_i$ matches all of $R'_i$ then return $M'_i$.
11.     let $B_i$ be the vertices of $R'_i$ left unmatched by $M'_i$.
12.     set $L_{i+1} = L_i \cup B_i$ and $R_{i+1} = V \setminus L_{i+1}$.
13.     $i = i + 1$.
14. end while
```

Figure 3.1: Our algorithm for computing a maximum cardinality popular matching.
We will show in the next section that our algorithm maintains the following invariants:

- $M_i$ is good with respect to $(L_i, R_i)$.
- $M'_i$ is good with respect to $(L'_i, R'_i)$.

Our initialization step in Fig. 3.1 sets $(A_0 \cup B_0, V \setminus (A_0 \cup B_0))$ as our left-right partition to begin with, where $A_0$ is the set of men unmatched in $S$ and $B_0$ is the set of women unmatched in $S$. At the start of the $i$-th iteration, we have a partition $(L_i, R_i)$ of $V$.

- If the matching $M_i$ that results from the proposal-disposal algorithm on $(L_i, R_i)$ matches all the vertices of $R_i$, then $M_i$ is the desired matching.
- Else let $A_i$ be the set of *men* in $R_i$ who are unmatched in $M_i$. We run the proposal-disposal algorithm on $(L_i \cup A_i, R_i \setminus A_i)$. If the resulting matching $M'_i$ matches all the vertices on the right, then $M'_i$ is the desired matching.
- Else let $B_i$ be the unmatched vertices on the right (all these vertices will be *women*). We set $L_{i+1} = L_i \cup B_i$ and $R_{i+1} = R_i \setminus B_i$; the next iteration begins.

**Lemma 6.** *For every i, the set $B_i \subseteq \mathcal{B}$.*

*Proof.* The set $B_i$ is the set of vertices of $R'_i = R_i \setminus A_i$ that are unmatched in $M'_i$. The matching $M'_i$ is the result of vertices in $L'_i = L_i \cup A_i$ proposing and vertices in $R'_i$ disposing. Note that every vertex of $R'_i$ that was matched in $M_i$ with vertices in $L_i$ proposing, will remain matched in $M'_i$ with $L'_i = L_i \cup A_i$ proposing to $R'_i = R_i \setminus A_i$.

Thus every *man* in $R_i$ who was matched in $M_i$ will remain matched in $M'_i$. Since we moved all the unmatched men of $R_i$ (this is the set $A_i$) away from $R_i$ to form $R'_i = R_i \setminus A_i$, every vertex of $R'_i$ that is unmatched in $M'_i$ has to be a *woman*. That is, the set $B_i$ of vertices of $R_i$ that are unmatched in $M'$, is a subset of $\mathcal{B}$. $\square$

**Termination of the algorithm.** We now show that the while loop in Algorithm 1 runs for at most $|\mathcal{B}|$ iterations. This implies that the running time of our algorithm is $O(mn)$ as it is easy to see that every iteration takes $O(m)$ time.

**Lemma 7.** *The number of while-loop iterations in Algorithm 1 is at most $|\mathcal{B}|$.*

*Proof.* To show that termination has to happen within the first $|\mathcal{B}|$ iterations is simple. This is because if termination does not happen in the $i$-th iteration, then $L_{i+1} \supset L_i$ because $B_i \neq \emptyset$ (otherwise termination would have happened in the $i$-th iteration). Once a woman moves to the left side of the graph, she never moves back to the right side again. Thus there is an iteration $k$, for some $1 \leq k \leq |\mathcal{B}|$, where either $M_k$ matches all the vertices of $R_k$ or $M'_k$ matches all the women in $R'_k$ (in other words, $M'_k$ matches all the vertices of $R'_k$). Thus the termination condition gets satisfied, thus the algorithm terminates in the $k$-th iteration, for some $k \leq |\mathcal{B}|$. $\square$

## 3.1 Correctness of Algorithm 1

We will now show that for every $1 \leq i \leq$ number of iterations in our algorithm, the matchings $M_i$ and $M'_i$ are good with respect to their left-right partitions. Note that for all the matchings $M_i$ and $M'_i$ computed in our algorithm, property (1) of goodness is obvious since these matchings are obtained by the proposal-disposal algorithm between the left side and the right side (see Claim 4). What we need to show now is that property (2) of goodness is also obeyed by them.

We know that $M_1$ is good with respect to $(L_1, R_1)$ (see Section 2.2). This is because $L_1$ is an independent set and so property (2) of goodness is vacuously true. The next lemma shows that $M'_1$ obeys property (2) of goodness.

**Lemma 8.** *If* $(a,b) \in L'_1 \times L'_1$, *then* $\mathsf{vote}_a(b, M'_1(a)) = -1$ *and* $\mathsf{vote}_b(a, M'_1(b)) = -1$.

*Proof.* Let $e = (a,b)$ be any edge in $L'_1 \times L'_1$. Since $L'_1 = A_0 \cup B_0 \cup A_1$ where $A_0 \cup B_0$ is an independent set, the vertex $a$ has to be in $A_1$. Observe that every vertex of $A_1$ will be matched in $M'_1$ by virtue of the fact that the other vertices in $L'_1$ comprise the set of vertices unmatched in any stable matching of $G$. It is easy to see that $a \in A_1$ gets a partner in $M'_1$ that is at least as good as $S(a)$, where $S$ is the stable matching that results from vertices in $\mathcal{A}$ proposing to vertices in $\mathcal{B}$. Recall that $B_0$ is the set of women unmatched in $S$, so $a$ regards $S(a)$ better than any neighbor in $B_0$. Thus $a$ prefers $M'_1(a)$ to all his neighbors in $B_0$, hence $\mathsf{vote}_a(b, M'_1(a)) = -1$.

Now we show that $\mathsf{vote}_b(a, M'_1(b)) = -1$. Recall that each man in $A_1$ was left *unmatched* in $M_1$: so $b \in B_0$ prefers $M_1(b)$ to all her neighbors in $A_1$. Observe that no vertex $b$ of $B_0$ gets dislodged from $M_1(b)$ (a man) by the presence of $A_1$ in $L'_1$ since vertices of $A_1$ propose to *women*. Thus $M'_1(b) = M_1(b)$ and so $\mathsf{vote}_b(a, M'_1(b)) = -1$. This finishes the proof of the lemma. □

This proves that $M'_1$ is good with respect to $(L'_1, R'_1)$. Now consider any $i \geq 2$. We assume by induction hypothesis on $i$ that the matching $M'_{i-1} \subseteq L'_{i-1} \times R'_{i-1}$ is good with respect to $(L'_{i-1}, R'_{i-1})$.

Lemma 9 shows that then $M_i \subseteq L_i \times R_i$ will be good with respect to $(L_i, R_i)$.

**Lemma 9.** *If* $(a,b) \in L_i \times L_i$, *then* $\mathsf{vote}_a(b, M_i(a)) = -1$ *and* $\mathsf{vote}_b(a, M_i(b)) = -1$.

*Proof.* The set $L_i = A_0 \cup B_0 \cup B_1 \cup \cdots \cup B_{i-1}$. Let $e = (a,b) \in L_i \times L_i$. So $a$ has to be in $A_0$ and $b \in B_0 \cup \cdots \cup B_{i-1}$. We need to show that every $a \in A_0$ prefers $M_i(a)$ to his neighbors in $B_0 \cup \cdots \cup B_{i-1}$ and every $b \in B_0 \cup \cdots \cup B_{i-1}$ prefers $M_i(b)$ to her neighbors in $A_0$.

By induction hypothesis, we know that $M'_{i-1}$ is good with respect to $(L'_{i-1}, R'_{i-1})$, where the set $L'_{i-1} = A_0 \cup B_0 \cup \cdots \cup B_{i-2} \cup A_{i-1}$. So for every edge $(a,b) \in L'_{i-1} \times L'_{i-1}$, we know that $a$ prefers $M'_{i-1}(a)$ to any neighbor $b$ in $B_0 \cup \cdots \cup B_{i-2}$. It is easy to see that any $a \in A_0$ gets at least as good as partner in $M_i$ as in $M'_{i-1}$ because $L_i = (L'_{i-1} \setminus A_{i-1}) \cup B_{i-1}$.

- The presence of $B_{i-1}$ in $L_i$ does not hurt the men in $A_0$ when they propose to women in $R_i$ because $B_{i-1}$ is the set of women who were *unmatched* on the right when then men in $A_0$ were proposing in $(L'_{i-1}, R'_{i-1})$.
- Also, the absence of $A_{i-1}$ on the left helps the men in $A_0$ as they are the only men proposing on the left now in $(L_i, R_i)$ in comparison with $(L'_{i-1}, R'_{i-1})$.

Thus for any $a \in A_0$ and $a$'s neighbor $b \in B_0 \cup \cdots \cup B_{i-2}$, $\mathsf{vote}_a(b, M_i(a)) = -1$. Also, for any $a \in A_0$ and neighbor $b \in B_{i-1}$, we know that $\mathsf{vote}_a(b, M'_{i-1}(a)) = -1$ since the vertices of $B_{i-1}$ were unmatched in $M'_{i-1}$, hence $\mathsf{vote}_a(b, M_i(a)) = -1$.

Now we show that for every $(a,b) \in L_i \times L_i$, the vertex $b$ also votes $-1$ for $a$ vs $M_i(b)$. First, there are no edges between $B_0$ and $A_0$. So $b \in B_1 \cup \cdots \cup B_{i-1}$. Each woman in $B_1 \cup \cdots \cup B_{i-1}$ is matched in any stable matching of $G$. Hence when they propose to men in $\mathcal{A} \setminus A_0$, each woman in $B_1 \cup \cdots \cup B_{i-1}$ gets matched to a man that she considers better than her neighbors in $A_0$. Thus for any $b \in B_1 \cup \cdots \cup B_{i-1}$ and $b$'s neighbor $a \in A_0$, $\mathsf{vote}_b(a, M_i(b)) = -1$. This finishes the proof of this lemma. □

Suppose $M_i$ does not match all the vertices in $R_i$, then we run the second stage of the $i$-th iteration, where all the men of $R_i$ who were left unmatched by $M_i$ (call this set $A_i$) are moved to the left. Thus $L'_i = L_i \cup A_i$. That is, $L'_i = A_0 \cup B_0 \cup \cdots \cup B_{i-1} \cup A_i$.

The proposal-disposal algorithm between $L'_i$ and $R'_i$ results in the matching $M'_i$. We will now show that property (2) of goodness also holds for $M'_i$. By induction hypothesis on $i$, we know that the matching $M'_{i-1}$ is good with respect to $(L'_{i-1}, R'_{i-1})$. The following claim will be helpful to us.

**Claim 5.** *The set $A_i \subseteq A_{i-1}$, where $A_{i-1}$ is the set of men in $R_{i-1}$ left unmatched in $M_{i-1}$.*

*Proof.* The set $A_{i-1}$ was the set of men in $R_{i-1}$ left unmatched in $M_{i-1}$. Note that every vertex that was matched in $R_{i-1}$ with $L_{i-1} = A_0 \cup B_0 \cup \cdots \cup B_{i-2}$ proposing, will remain matched with $L_{i-1} \cup A_{i-1}$ proposing to $R_{i-1} \setminus A_{i-1}$. Thus every *man* in $R_{i-1}$ who was matched $M_{i-1}$ will remain matched in $M'_{i-1}$ with the women in $B_0 \cup \cdots \cup B_{i-2}$ proposing on the left. At the start of the $i$-th iteration, the set $A_{i-1}$ goes back to the right and the set $B_{i-1}$ moves to the left. With the women in $B_0 \cup \cdots \cup B_{i-1}$ proposing, all the men who were matched in the second stage of the previous iteration, continue to remain matched and some of $A_{i-1}$ also possibly get matched. So the set of men in $R_i$ who are unmatched in $M_i$ is a subset of $A_{i-1}$, that is, $A_i \subseteq A_{i-1}$. □

We will now show that for any $(a,b) \in L'_i \times L'_i$, $\mathsf{vote}_a(b, M'_i(a)) = -1$ and $\mathsf{vote}_b(a, M'_i(b)) = -1$ in Lemmas 10 and 11, respectively.

**Lemma 10.** *If $(a,b) \in L'_i \times L'_i$, then $\mathsf{vote}_a(b, M'_i(a)) = -1$.*

*Proof.* We know from Claim 5 that $A_i \subseteq A_{i-1}$. Now $B_{i-1}$ is the set of women left *unmatched* in $R'_{i-1}$ when $A_0 \cup A_{i-1}$ was proposing on the left in the second stage of the $(i-1)$-th iteration. Hence each man $a \in A_0 \cup A_{i-1}$ prefers $M'_{i-1}(a)$ to any neighbor $b \in B_{i-1}$. Each man in $A_0 \cup A_i$ gets at least as good a neighbor in $M'_i$ as in $M'_{i-1}$ because

- there are fewer men proposing now than in the second stage of the $(i-1)$-th iteration as $A_0 \cup A_i \subseteq A_0 \cup A_{i-1}$ and
- it is only the unmatched women who moved away from $R'_{i-1}$. Hence all women who belong to $\{M'_{i-1}(a) : a \in A_0 \cup A_{i-1}\}$ are still present in $R'_i$ for $A_0 \cup A_i$ to propose to.

We know from the induction hypothesis that $M'_{i-1}$ is good with respect to $(L'_{i-1}, R'_{i-1})$. Hence each $a \in A_0 \cup A_{i-1}$ prefers $M'_{i-1}(a)$ to any neighbor in $B_0 \cup \cdots \cup B_{i-2}$. Also, we just argued that $a \in A_0 \cup A_{i-1}$ prefers $M'_{i-1}(a)$ to any neighbor in $B_{i-1}$. Since $A_i \subseteq A_{i-1}$ and because $M'_i(a)$ is at least as good as $M'_{i-1}(a)$ for all $a \in A_0 \cup A_i$, it follows that $\mathsf{vote}_a(b, M'_i(a)) = -1$ for any edge $(a,b)$ where $a \in A_0 \cup A_i$ and $b \in B_0 \cup \cdots \cup B_{i-2} \cup B_{i-1}$. □

**Lemma 11.** *If $(a,b) \in L'_i \times L'_i$, then $\mathsf{vote}_b(a, M'_i(b)) = -1$.*

*Proof.* Since $L'_i = A_0 \cup B_0 \cup \cdots \cup B_{i-1} \cup A_i$, for any $(a,b) \in L'_i \times L'_i$, the vertex $a \in A_0 \cup A_i$.
*Case 1*: Suppose $a \in A_i$. Since $A_i$ is the set of men in $R_i$ who are *unmatched* in $M_i$, it follows that each of $b \in B_0 \cup \cdots \cup B_{i-1}$ prefers $M_i(b)$ to any neighbor in $A_i$. Also for any $b \in B_0 \cup \cdots \cup B_{i-1}$, we have $M'_i(b) = M_i(b)$, since it is only *unmatched men* that moved from $R_i$ to the left side to form $L'_i$. Thus $\mathsf{vote}_b(a, M'_i(b)) = -1$.

*Case 2*: Suppose $a \in A_0$. Consider any edge between a man in $A_0$ and a woman $b \in B_0 \cup \cdots \cup B_{i-1}$. In the first place, $b$ has to be in $B_1 \cup \cdots \cup B_{i-1}$ since $A_0 \cup B_0$ is an independent set. Every $b \in B_1 \cup \cdots \cup B_{i-1}$ prefers her partner $M'_i(b)$ to any neighbor in $A_0$, since $A_0$ is the set of unmatched men in any stable matching of $G$. Thus $\mathsf{vote}_b(a, M'_i(b)) = -1$.

Hence for any $b \in B_0 \cup \cdots \cup B_{i-1}$, and any neighbor $a \in A_0 \cup A_i$, we have $\mathsf{vote}_b(a, M'_i(b)) = -1$. $\square$

Thus property (2) of goodness is true for $M'_i$. We have thus shown that for every $i$, where $1 \leq i \leq$ number of iterations in our algorithm, $M_i$ is good with respect to $(L_i, R_i)$ and $M'_i$ is good with respect to $(L'_i, R'_i)$. Thus as soon as we find an $M_i$ or an $M'_i$ that matches all the vertices on the right, we have a good matching that matches all the vertices on the right.

We know by Lemma 7 that within the first $|\mathcal{B}|$ iterations of the while loop, there is an iteration $k$ such that either $M_k$ or $M'_k$ matches all the vertices on the right. Thus Algorithm 3.1 always returns a maximum cardinality popular matching (by Theorem 2). This completes the proof of correctness of our algorithm. Since the running time of Algorithm 3.1 is $O(mn)$, Theorem 1 stated in Section 1 follows.

## 4 Conclusions

We showed that a maximum cardinality popular matching in $G = (\mathcal{A} \cup \mathcal{B}, E)$ with strict preference lists can be computed in $O(mn)$ time, where $m$ is the number of edges in $G$ and $n$ is the number of vertices in $G$. Our algorithm computes a matching $M$ and a partition $(L, R)$ of $\mathcal{A} \cup \mathcal{B}$ such that $M$ is *good* with respect to $(L, R)$ and $M$ matches all the vertices in $R$. Our main theorem (Theorem 2) shows that such a matching is a maximum cardinality popular matching in $G$. The existence of such a matching also implies (by Corollary 1) that the *every* maximum cardinality popular matching in $G$ matches the same vertices.

An open problem is the complexity of determining if a non-bipartite graph $G = (V, E)$ with strict preference lists admits a popular matching or not. Note that there is a polynomial time algorithm [6] to decide if such an input (also known as the roommates problem) admits a stable matching or not. Any stable matching is popular but it could very well be the case that $G$ has no stable matching but it has a popular matching (a simple example of such an instance can be found in [2]). Another open problem is the complexity of finding a maximum cardinality popular matching in a roommates problem that admits a popular matching. Note that Theorem 2 holds for non-bipartite graphs too, however non-bipartite graphs need not always admit such a matching.

# Bibliography

[1] D.J. Abraham, R.W. Irving, T. Kavitha and K. Mehlhorn. Popular matchings. SIAM Journal on Computing, Vol.37, No.4, pp. 1030–1045, 2007.

[2] P. Biro, R. W. Irving, D. F. Manlove. Popular matchings in the Marriage and Roommates problems Proceedings of the seventh International Conference on Algorithms and Complexity (CIAC), 97-108, 2010.

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

[4] P. Gardenfors. *Match making: assignments based on bilateral preferences*. Behavioural Sciences, 20:166–173, 1975.

[5] D. Gusfield and R.W. Irving. *The Stable Marriage Problem: Structure and Algorithms*. MIT Press, 1989.

[6] R.W. Irving. *An Efficient Algorithm for the "Stable Roommates" Problem.* Journal of Algorithms, 6: 577-595, 1985.

[7] T. Kavitha, J. Mestre, and M. Nasre. Popular mixed matchings. In *Proceedings of the 36th International Colloquium on Automata, Languages and Programming*, pages 574–584, 2009.

[8] T. Kavitha and M. Nasre. Note: Optimal popular matchings. *Discrete Applied Mathematics*, 157(14):3181–3186, 2009.

[9] D. E. Knuth. Mariages Stables. Les Presses de L'Université de Montreal, 1976.

[10] M. Mahdian. Random popular matchings. In *Proceedings of the 7th ACM Conference on Electronic-Commerce*, pages 238–242, 2006.

[11] D.F. Manlove and C. Sng. Popular matchings in the capacitated house allocation problem. In *Proceedings of ESA 2006, the 14th Annual European Symposium on Algorithms*, (LNCS 4168), pages 492-503, Springer-Verlag, 2006.

[12] M. McCutchen. The least-unpopularity-factor and least-unpopularity-margin criteria for matching problems with one-sided preferences. In *Proceedings of LATIN 2008, the 8th Latin American Theoretical Informatics Symposium.*

[13] E. McDermid and R. W. Irving. Popular matchings: Structure and algorithms. In *Proceedings of 15th Annual International Computing and Combinatorics Conference*, pages 506–515, 2009.

[14] J. Mestre. *Weighted popular matchings.* In *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming*, (LNCS 4051), pages 715–726, 2006.

---

| | | |
|---|---|---|
| MPI-I-2010-RG1-001 | M. Suda, C. Weidenbach, P. Wischnewski | On the saturation of YAGO |
| MPI-I-2010-5-007 | J. Hoffart, F.M. Suchanek, K. Berberich, G. Weikum | YAGO2: a spatially and temporally enhanced knowledge base from Wikipedia |
| MPI-I-2010-5-006 | A. Broschart, R. Schenkel | Real-time text queries with tunable term pair indexes |
| MPI-I-2010-5-005 | S. Seufert, S. Bedathur, J. Mestre, G. Weikum | Bonsai: Growing Interesting Small Trees |
| MPI-I-2010-5-003 | A. Anand, S. Bedathur, K. Berberich, R. Schenkel | Efficient temporal keyword queries over versioned text |
| MPI-I-2010-5-002 | M. Theobald, M. Sozio, F. Suchanek, N. Nakashole | URDF: Efficient Reasoning in Uncertain RDF Knowledge Bases with Soft and Hard Rules |
| MPI-I-2010-5-001 | K. Berberich, S. Bedathur, O. Alonso, G. Weikum | A language modeling approach for temporal information needs |
| MPI-I-2009-RG1-005 | M. Horbach, C. Weidenbach | Superposition for fixed domains |
| MPI-I-2009-RG1-004 | M. Horbach, C. Weidenbach | Decidability results for saturation-based model building |
| MPI-I-2009-RG1-002 | P. Wischnewski, C. Weidenbach | Contextual rewriting |
| MPI-I-2009-RG1-001 | M. Horbach, C. Weidenbach | Deciding the inductive validity of $\forall\exists^*$ queries |
| MPI-I-2009-5-007 | G. Kasneci, G. Weikum, S. Elbassuoni | MING: Mining Informative Entity-Relationship Subgraphs |
| MPI-I-2009-5-006 | S. Bedathur, K. Berberich, J. Dittrich, N. Mamoulis, G. Weikum | Scalable phrase mining for ad-hoc text analytics |
| MPI-I-2009-5-005 | G. de Melo, G. Weikum | Towards a Universal Wordnet by learning from combined evidenc |
| MPI-I-2009-5-004 | N. Preda, F.M. Suchanek, G. Kasneci, T. Neumann, G. Weikum | Coupling knowledge bases and web services for active knowledge |
| MPI-I-2009-5-003 | T. Neumann, G. Weikum | The RDF-3X engine for scalable management of RDF data |
| MPI-I-2009-5-002 | M. Ramanath, K.S. Kumar, G. Ifrim | Generating concise and readable summaries of XML documents |
| MPI-I-2009-4-006 | C. Stoll | Optical reconstruction of detailed animatable human body models |
| MPI-I-2009-4-005 | A. Berner, M. Bokeloh, M. Wand, A. Schilling, H. Seidel | Generalized intrinsic symmetry detection |
| MPI-I-2009-4-004 | V. Havran, J. Zajac, J. Drahokoupil, H. Seidel | MPI Informatics building model as data for your research |
| MPI-I-2009-4-003 | M. Fuchs, T. Chen, O. Wang, R. Raskar, H.P.A. Lensch, H. Seidel | A shaped temporal filter camera |
| MPI-I-2009-4-002 | A. Tevs, M. Wand, I. Ihrke, H. Seidel | A Bayesian approach to manifold topology reconstruction |
| MPI-I-2009-4-001 | M.B. Hullin, B. Ajdin, J. Hanika, H. Seidel, J. Kautz, H.P.A. Lensch | Acquisition and analysis of bispectral bidirectional reflectance distribution functions |
| MPI-I-2008-RG1-001 | A. Fietzke, C. Weidenbach | Labelled splitting |
| MPI-I-2008-5-004 | F. Suchanek, M. Sozio, G. Weikum | SOFI: a self-organizing framework for information extraction |
| MPI-I-2008-5-003 | G. de Melo, F.M. Suchanek, A. Pease | Integrating Yago into the suggested upper merged ontology |
| MPI-I-2008-5-002 | T. Neumann, G. Moerkotte | Single phase construction of optimal DAG-structured QEPs |
| MPI-I-2008-5-001 | G. Kasneci, M. Ramanath, M. Sozio, F.M. Suchanek, G. Weikum | STAR: Steiner tree approximation in relationship-graphs |