Egerváry Research Group
on Combinatorial Optimization

Technical reportS

TR-2016-16. Published by the Egerváry Research Group, Pázmány P. sétány 1/C,
H–1117, Budapest, Hungary. Web site: www.cs.elte.hu/egres . ISSN 1587–4451.

Finding strongly popular matchings in certain bipartite preference systems

Tamás Király and Zsuzsa Mészáros-Karkus

November 2016

Finding strongly popular matchings in certain bipartite preference systems

Tamás Király* and Zsuzsa Mészáros-Karkus**

Abstract

The computational complexity of the popular matching problem in bipartite preference systems with ties depends greatly on the structure of ties. If one side has strict preferences while nodes on the other side are indifferent (but prefer to be matched), then a popular matching can be found in polynomial time [Cseh, Huang, Kavitha, 2015]. However, as the same paper points out, the problem becomes NP-complete if one side has strict preferences while the other side can have both indifferent nodes and nodes with strict preferences. We show that the problem of finding a *strongly* popular matching is polynomial-time solvable even in the latter case.

1 Introduction

A *bipartite preference system with ties* consists of a bipartite multigraph $G = (S, T; E)$ and partial orders $\preceq_v$ on the edges incident to v, for every node $v \in S \cup T$. Given a bipartite preference system with ties, a node *prefers* a matching M_1 to a matching M_2 if either it is matched in M_1 but not in M_2, or it is matched by a better edge in M_1 than in M_2. A matching M_1 is *more popular* than matching M_2 if the number of nodes preferring M_1 to M_2 is strictly larger than the number of nodes preferring M_2 to M_1. This relation is not transitive; it is possible that M_1 is more popular than M_2, M_2 is more popular than M_3, and M_3 is more popular than M_1 [2]. A matching M is *popular* if no matching is more popular than M, and it is *strongly popular* if M is more popular than any other matching. These notions were first introduced by Gärdenfors [8], who showed that *a)* every strongly popular matching is stable and *b)* in case of no ties, all stable matchings are popular.

Obviously, an instance cannot have two strongly popular matchings, because both of them would be more popular than the other, which is impossible. Furthermore, a strongly popular matching must be a unique popular matching; however, there are

*MTA-ELTE Egerváry Research Group, Department of Operations Research, Eötvös University, Pázmány Péter sétány 1/C, Budapest, Hungary, H-1117. Email: tkiraly@cs.elte.hu

**Department of Operations Research, Eötvös University, Pázmány Péter sétány 1/C, Budapest, Hungary, H-1117. Email: karkuszsuzsi@gmail.com

instances where the popular matching is unique but it is not strongly popular (see the full version of [2] for an example).

Algorithmic questions about popular matchings have generated a lot of interest lately; see Section 1.1 for a short summary of recent results. Here we just mention that for any preference system with ties (even non-bipartite), it can be decided in polynomial time if a given matching is popular or strongly popular [2]. This means that the decision problem for popular matchings is in the complexity class NP, while the decision problem for strongly popular matchings is in the lesser-known complexity class UP (Unambiguous Polynomial-time). The latter class, introduced by Valiant [15], consists of the decision problems solvable by an NP machine such that all witnesses are rejected in a "no" instance, while exactly one witness is accepted in a "yes" instance. The strongly popular matching problem belongs to this class because in a "yes" instance there is a unique strongly popular matching and it can be verified in polynomial time.

In this paper we consider bipartite preference systems with two types of nodes: *nodes with strict preferences*, where the preference order $\preceq_v$ is a total order, and *indifferent nodes*, where every incident edge is equally good (but who still prefer to be matched). If all nodes have strict preferences, then every stable matching is popular [8]. On one hand, this implies that there always exists a popular matching and one can be found using the well-known Gale-Shapley algorithm [7]. On the other hand, we can decide if a strongly popular matching exists by finding an arbitrary stable matching and checking whether it is strongly popular (this also works in non-bipartite preference systems without ties [2]).

The problems become more complex when indifferent nodes are also allowed on one of the sides. If nodes on one side have strict preferences while those on the other side are all indifferent, then the existence of a popular matching can still be decided in polynomial time, as shown by Cseh, Huang, and Kavitha [4]. However, they also showed that the problem becomes NP-complete if one side has strict preferences while the other side may feature both indifferent nodes and nodes with strict preferences; see the full version of [4] and [5] for proofs.

The main result of the present paper is that the existence of a *strongly* popular matching can be decided in polynomial time even in the latter case.

Theorem 1.1. *Given a bipartite preference system $(G = (S, T; E), \preceq)$ where nodes in S have strict preferences and each node in T is either indifferent or has strict preferences, it can be decided in polynomial time if there is a strongly popular matching.*

The algorithm succesively finds edges that *cannot* be in a strongly popular matching or *must* be in any strongly popular matching, and also maintains a directed graph related to the possible structure of the strongly popular matching. The set of possible candidates keeps shrinking until, at the end, we can either conclude that there is no strongly popular matching, or we have exactly one candidate. We can then check in polynomial time whether this matching is strongly popular.

1.1 Other related work

There is a lot of ongoing research about the computational complexity of the popular matching problem. For bipartite preference systems with no ties, Huang and Kavitha [9] showed that a maximum size popular matching can be found in polynomial time, while Cseh and Kavitha [6] gave an algorithm for deciding if a given edge belongs to a popular matching. The former result can also be extended to the Hospitals-Residents problem, where more than one residents can be matched to a hospital [3, 14]. On the other hand, the complexity of deciding the existence of a popular matching in a non-bipartite preference system without ties is still open. Huang and Kavitha [10] introduced the notion of unpopularity factor, and showed that, for any positive ε, it is NP-hard to compute a matching with unpopularity factor within $\frac{4}{3} - \varepsilon$ of optimal.

Several recent results concern a slightly different, one-sided model (also called the House Allocation model), where one side has preference lists, while nodes on the other side do not vote at all and do not prefer to be matched. Abraham et al. [1] gave a polynomial-time algorithm for finding a popular matching in this model. If the preference lists are strict, then optimal popular matchings can also be found for various notions of optimality [12, 13].

2 Proof of the main theorem

In this section we prove Theorem 1.1. We are given a bipartite multigraph $G = (S, T; E)$, and the node set T is partitioned into two parts, T_P and T_I. The nodes in $S \cup T_P$ have strict preference orders $\preceq_v$ over their incident edges, while the nodes in T_I are indifferent but prefer to be matched. We give a polynomial-time algorithm which decides if the instance admits a strongly popular matching (SPM for short).

2.1 Preliminaries

Before going into the details, we give an overview of the main ideas of the proof. During the algorithm, we modify the instance using the following two operations.

1. We remove edges that cannot appear in an SPM of the current instance,
2. We fix edges that must belong to the SPM of the current instance (if it exists). Fixed edges are removed together with their two endnodes. The set of fixed edges is denoted by F.

Let $G^k = (S^k, T^k; E^k)$ be the current instance after performing k of the above operations, and let F be the set of edges fixed so far.

Lemma 2.1. *If the original instance has an SPM M, then $F \subseteq M$, and $M \setminus F$ is an SPM of G^k.*

Proof. We prove by induction on k; let G^{k-1} be the instance before the last operation. If the last operation was the removal of an edge st, then, by induction, M contains F, $M \setminus F$ is an SPM of G^{k-1}, and $st \notin M$. Thus $M \setminus F$ is an SPM of G^k.

If we fixed an edge st in the last operation, then $M \setminus (F - st)$ is an SPM of G^{k-1} by induction, and $st \in M \setminus (F - st)$ because we only fix edges with this property. This implies that $st \in M$, and therefore $M \setminus F$ is an SPM of G^k. $\square$

Note that it is possible that G^k has an SPM even if G does not have one. However, this is not a problem: if we eventually obtain an empty graph by repeating the operations, then F is the only candidate for an SPM, and we can check in polynomial time if it is an SPM of G or not. On the other hand, if we obtain a graph G^k that has no SPM, then G also has none.

An edge $st \in E$ is called a *blocking edge* with respect to a matching M if both s and t prefer the edge st to their partner in the matching (this includes the case when $t \in T_I$ and it is unmatched). If M is an SPM, then there is no blocking edge with respect to M; indeed, if M' is the matching obtained from M by adding a blocking edge st and removing the original edges incident to s and t, then M is not more popular than M'. We will use the term "blocking edge" in another sense for parallel edges: if e and e' are parallel edges and one endpoint prefers e to e', then e blocks any matching M containing e', since $M - e' + e$ is at least as popular as M.

In addition to blocking edges, we will use alternating paths and cycles to show that certain matchings cannot be strongly popular. Given a matching M and an alternating path or cycle w.r.t. M, let M' be the matching obtained from M by exchanging along the path or cycle (if we exchange along a path whose first or last edge is not in M, then we also remove the edge of M covering the corresponding endpoint of the path). If we can show that M' is at least as popular as M, then M is not an SPM.

2.2 First phase of the algorithm

The algorithm starts with a first phase that is reminiscent of the first phase of Irving's algorithm for the stable roommates problem [11]. We repeat the following steps.

- From every node in $S \cup T_P$ we draw a directed edge to its first choice.
- If there is a directed edge to a node $v \in S \cup T_P$, then we delete the edges incident to v which are worse according to $\preceq_v$ than the incoming directed edge. We also delete the edges parallel to the directed edge. If we deleted a node's first choice, then we draw a directed edge to its first choice among its remaining neighbors.

Claim 2.2. *The deleted edges cannot belong to an SPM.*

Proof. Suppose that uv belongs to an SPM and it was deleted because of a directed edge wv. Then wv is a blocking edge with respect to the SPM, a contradiction. $\square$

Claim 2.3. *If at some point there is only one directed edge st entering a node in T_I, then st belongs to the SPM if there is one.*

Proof. Suppose that the SPM M does not contain st; then t has to be matched to a node $u \neq s$, otherwise st would be blocking. Consider the path that starts with s and alternates between directed edges and edges of M. (The first edge is st, the second is

tu.) If we reach a node $t' \in T_I$, then by exchanging along the path we get a matching which is as popular as M: the nodes of S on the path all get a better partner, while the only nodes that may prefer M are the nodes of T in the path except for t and t', and the partners of t' and s in M. This contradicts the assumption that M is an SPM.

If we return to s, then exchanging along the cycle yields a matching at least as popular as M. See Figure 1 for an illustration of both cases. □

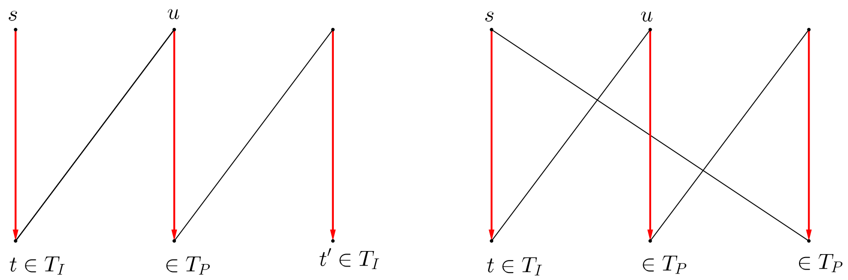

Figure 1: The black edges belong to the SPM.

Claim 2.3 means that we can fix the edge st to be in F, and delete s and t from the graph.

Claim 2.4. *If at some point there is a node $t \in T_I$ which is not an endpoint of a directed edge but there is an edge st which has not been deleted, then an edge su cannot belong to the SPM if s prefers st over su.*

Proof. Suppose that su is in the SPM. The node t has to be matched to some node v, since otherwise st would be blocking. Consider the path which starts with us, st, tv and then alternates between directed edges and edges of the SPM. Similarly to the proof of Claim 2.3, if we reach a node in T_I, then exchanging the edges along the path yields a matching preferred by the same number of nodes as the original SPM, while if the path returns to u, then by exchanging along the cycle we get a new matching that is as popular as the SPM, a contradiction. □

It follows from the claim that we can delete such edges su, and continue phase 1. We can also delete the nodes in $S \cup T_P$ which become isolated. If the graph becomes empty at the end of phase 1, then we can check whether the set F of fixed edges is an SPM in the original graph G, so we are done by Lemma 2.1. Otherwise we proceed to phase 2, which is described below.

2.3 Second phase of the algorithm

Let D' denote the directed graph obtained at the end of phase 1, and let G' be the bipartite graph consisting of all nodes and edges that have not been deleted in the first phase. D' can have three types of components:

- directed cycles;
- in-arborescences with a root-node (sink) in T_I having in-degree at least 2. The other nodes of the arborescence are in $S \cup T_P$ and they have out-degree 1 and in-degree at most 1. Therefore, each arborescence consists of disjoint directed paths leading to the root-node; the first nodes of these paths are called *leaves*.
- isolated nodes that are in T_I (note that these nodes are not isolated in G').

Let T_1 denote the nodes in T_I which are root-nodes of one of the arborescences, and let T_2 denote the isolated nodes in D'.

Lemma 2.5. *If uv is an edge in $G'[S \cup T_P \cup T_1]$ and it is not a directed edge in any direction, then uv cannot belong to the SPM.*

Proof. Suppose uv is in the SPM M. If $u \in T_1$, then there is a directed edge su in D', for some $s \neq v$. Consider the path starting with su, uv and then alternating between directed edges and edges of M. Similarly to the proof of Claim 2.3, we either reach a node in T_I or return to s, and exchanging along the obtained path or cycle yields a matching that is preferred by at least as many nodes as the number of nodes that prefer M.

Now consider the case where $u \in T_P$. Consider the path starting with vu and then alternating between directed edges and edges of M. If we return to v without reaching a node in T_1, then exchanging along the cycle yields a matching that is at least as popular as M. If we reach a node in T_1, then there is another directed edge pointing to this node, which we add to the path. Let this path be denoted by P. We continue P from v with edges alternating between directed edges and edges of M. If we reach a node in T_I, then exchanging along the path yields a matching that is at least as popular as M; see Figure 2 for an illustration of this case. If we return to a node in P, then, again, exchanging along the obtained cycle yields a matching at least as popular as M. (One of the endpoints of each edge in M is better off with the new matching except for maybe one edge, but u and v are both better off.) $\square$

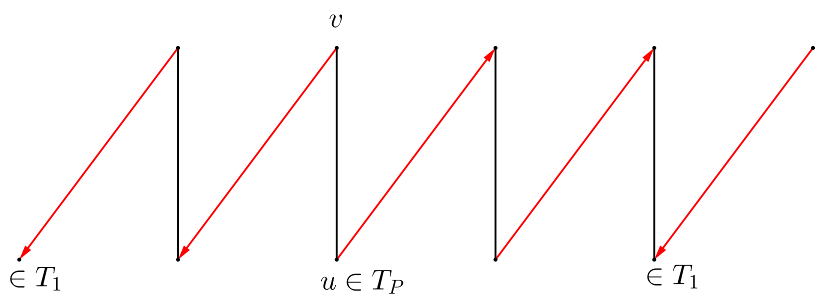

Figure 2: The black edges belong to the SPM.

The lemma implies that only the edges of D' and edges of G' with one endpoint in T_2 can belong to the SPM. Therefore we can delete the other edges. From Claim 2.4, it follows that for every node $s \in S$ there can be only one edge between s and T_2.

Lemma 2.6. *If there is a cycle of length more than 2 in D' then there is no SPM.*

Proof. Suppose that there is an SPM M. If one of the nodes v of the cycle is matched to a node $t \in T_2$ in M, then node v prefers its predecessor u in the cycle (because of Claim 2.4), and therefore uv is a blocking edge with respect to M.

If every node of the cycle is matched along the cycle, then we can exchange along the cycle to get a matching at least as popular as M. $\square$

A cycle of length 2 in D' corresponds to a single edge that must belong to the SPM, so we can fix these edges and delete their endpoints.

Claim 2.7. *In the SPM, only the leaves (i.e. the nodes of in-degree 0 and out-degree 1 in D') can be matched to nodes in T_2.*

Proof. Suppose u is matched to $t \in T_2$ in the SPM and u is not a leaf; therefore, there is a node v such that vu is in the arborescence. Because of Claim 2.4, vu is a blocking edge. $\square$

By the claim, we can delete the edges between T_2 and any node which is not a leaf.

Claim 2.8. *Every leaf is matched in the SPM.*

Proof. Let M be the SPM, and suppose there is a leaf $s \in S$ that is not matched in M. The other nodes of the branch containing s, except for the root, must be matched along the branch. By exchanging the edges along the branch such that the edge incident to s and the edge incident to the root belong to the new matching, we obtain a matching that is as popular as M.

Now suppose there is an unmatched leaf $t \in T$. Again, the other nodes of the branch must be matched along the branch, and now the root also has to be matched in this branch, otherwise there is a blocking edge. We exchange the edges along the branch and add to the matching another edge pointing to the root (here we use the property that the in-degree of the root is at least 2). If the tail of this edge is covered by M, then we remove the edge covering it from the matching. It is easy to check that the new matching is at least as popular as M. $\square$

If there is an arborescence with all leaves in T, then all of its nodes have to be matched along the arborescence, and from the above claim all of its nodes have to be matched. But the arborescence has an odd number of nodes, therefore there cannot be an SPM.

If an arborescence has only one leaf in S, then its nodes have to be matched along the arborescence, and there is a unique way to match them (see Figure 3). Therefore we can fix these edges and delete the arborescence.

Claim 2.9. *If a node $t \in T_2$ has degree 1 in G', then it has to be matched in the SPM.*

Proof. Suppose that t is not matched in the SPM. Let ts be the only edge incident to t in G', and let r be the root of the arborescence that s belongs to. By Claim 2.7, s is a leaf of this arborescence. If r is not matched along the branch of s, then s cannot be

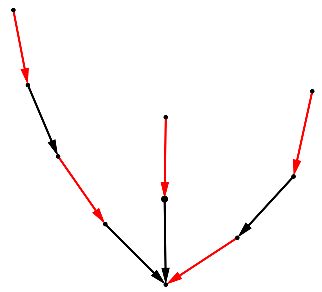

Figure 3: The red edges give the only possible SPM.

matched, and therefore st is blocking. If r is matched along the branch of s, then we exchange the edges along the branch and add ts and vr to the new matching, where vr is an edge of the arborescence from another branch; we also remove the original matching edge covering v. The new matching is at least as popular as the SPM, a contradiction. □

By the claim, if a node $t \in T_2$ has a single neighbor s in G', then we can fix ts and every second edge of the branch of s, and delete this branch and t. Suppose that this creates an arborescence with a single branch; then the original arborescence had two leaves, both in S (as we have already removed arborescences with only one leaf in S), and since the root cannot be matched on the branch of s, there is a unique way to match the whole arborescence. So in this case we can fix the matching on both branches and remove the whole arborescence, maintaining the property that every arborescence has at least two branches.

After performing all the above operations, the following hold.

- there are no parallel edges in G';
- every arborescence has at least two leaves in S;
- every leaf in S has at most one neighbor in T_2;
- every node in T_2 has degree at least 2;
- every node of each arborescence is matched in the SPM.

These properties can be satisfied only if every arborescence has exactly 2 leaves in S, every leaf in S has exactly one neighbor in T_2, and every node in T_2 has degree 2. This means that the graph contains a cycle if it is nonempty, and, in addition, every second edge in this cycle must be in the SPM. However, we can exchange along the cycle to get a new matching at least as popular as the SPM, which is a contradiction. We can conclude that the remaining graph is empty, which means that the only possible candidate for an SPM is F, i.e. the set of edges that we fixed. We can check in polynomial time if this is an SPM or not. □

Remark 2.10. It is easy to see that we can apply the above algorithm with slight modifications to the case where nodes in S have strict preferences, and the preference lists of nodes in T can contain one tie, of arbitrary length, at the end.

3 Conclusion

We proved that in case of strict preferences on one side and both strict preferences and indifference on the other side, the existence of a strongly popular matching can be decided in polynomial time. This is a clear indication that the strongly popular matching problem is significantly easier than the popular matching problem. It seems to be difficult to complement this with hardness results; as mentioned in the introduction, the strongly popular matching problem is in the complexity class UP, for which no complete problems are known. Therefore the more promising direction is to prove polynomial-time solvability for other types of preference systems. In praticular, the decision problem for strongly popular matchings is open in the following two cases:

- bipartite preference systems with strict preference and indifference allowed on both sides,
- bipartite preference systems with strict preferences on one side, and arbitrary preferences on the other side.

Our techniques do not seem to extend easily to these problems.

Acknowledgement

The research was supported by the Hungarian National Research, Development and Innovation Office – NKFIH, grant number K120254.

References

[1] D.J. Abraham, R.W. Irving, T. Kavitha, K. Mehlhorn, *Popular matchings*, SIAM Journal on Computing 37 (2007), 1030–1045.

[2] P. Biró, R. W. Irving, and D. F. Manlove, *Popular matchings in the marriage and roommates problems*, In Proceedings of the 7th International Conference on Algorithms and Complexity, volume 6078 of Lecture Notes in Computer Science (2010), 97–108. Full version: Technical Report TR-2009-306, University of Glasgow, Department of Computing Science.

[3] F. Brandl, T. Kavitha, *Popular Matchings with Multiple Partners*, arXiv: 1609.07531 (2016).

[4] Á. Cseh, C.-C. Huang, and T. Kavitha, *Popular matchings with two-sided preferences and one-sided ties*, In M. M. Halldórsson, K. Iwama, N. Kobayashi,

and B. Speckmann, editors, Automata, Languages, and Programming, volume 9134 of Lecture Notes in Computer Science (2015), 367–379. Full version: arXiv: 1603.07168

[5] Á. Cseh, *Complexity and algorithms in matching problems under preferences*, Ph.D. Thesis, TU Berlin (2016).

[6] Á. Cseh, T. Kavitha, *Popular Edges and Dominant Matchings*, Integer Programming and Combinatorial Optimization, Volume 9682 of the series Lecture Notes in Computer Science (2016), 138–151.

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

[8] P. Gärdenfors, *Match making: assignments based on bilateral preferences*, Behavioural Science 20 (1975), 166–173.

[9] C.-C. Huang, T. Kavitha, *Popular Matchings in the Stable Marriage Problem*, Automata, Languages and Programming, Volume 6755 of the series Lecture Notes in Computer Science (2011), 666–677.

[10] C.-C. Huang, T. Kavitha, *Near-Popular Matchings in the Roommates Problem*, SIAM J. Discrete Math. 27 (2013), 43–62.

[11] R.W. Irving, *An efficient algorithm for the "stable roommates" problem*, Journal of Algorithms 6 (1985), 577–595.

[12] T. Kavitha, M. Nasre, *Optimal popular matchings*, Discrete Applied Mathematics 157 (2009), 3181–3186.

[13] E. McDermid, R.W. Irving, *Popular matchings: Structure and algorithms*, In Proceedings of COCOON 2009: the 15th Annual International Computing and Combinatorics Conference, volume 5609 of Lecture Notes in Computer Science (2009), 506–515.

[14] M. Nasre, A. Rawat, *Popularity in the generalized Hospital Residents Setting*, arXiv:1609.07650 (2016).

[15] L. G. Valiant, *Relative complexity of checking and evaluating*, Information Processing Letters 5 (1976), 20–23.