Theoretical Computer Science 412 (2011) 2679–2690

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Popular mixed matchings[☆]

Telikepalli Kavitha [a,1], Julián Mestre [b,*], Meghana Nasre [a]

[a] *Indian Institute of Science, Bangalore, India*

[b] *Max-Plank-Institut für Informatik, Saarbrücken, Germany*

ARTICLE INFO

Keywords:
Mixed matchings
Popular matchings
Stable assignments

ABSTRACT

We study the problem of matching applicants to jobs under one-sided preferences; that is, each applicant ranks a non-empty subset of jobs under an order of preference, possibly involving ties. A matching M is said to be *more popular* than T if the applicants that prefer M to T outnumber those that prefer T to M. A matching is said to be *popular* if there is no matching more popular than it. Equivalently, a matching M is popular if $\phi(M, T) \geq \phi(T, M)$ for all matchings T, where $\phi(X, Y)$ is the number of applicants that prefer X to Y.

Previously studied solution concepts based on the popularity criterion are either not guaranteed to exist for every instance (e.g., popular matchings) or are NP-hard to compute (e.g., least unpopular matchings). This paper addresses this issue by considering mixed matchings. A *mixed matching* is simply a probability distribution over matchings in the input graph. The function ϕ that compares two matchings generalizes in a natural manner to mixed matchings by taking expectation. A mixed matching P is popular if $\phi(P, Q) \geq \phi(Q, P)$ for all mixed matchings Q.

We show that popular mixed matchings *always* exist and we design polynomial time algorithms for finding them. Then we study their efficiency and give tight bounds on the price of anarchy and price of stability of the popular matching problem.

1. Introduction

We study the problem of matching a set of applicants $\mathcal{A}$ to a set of jobs $\mathcal{J}$ under one-sided preferences. More formally, the input consists of a bipartite graph $G = (\mathcal{A}, \mathcal{J}, E)$ and a rank function $r : E \to \mathbb{Z}$ that captures applicant preferences over the jobs. Given two jobs i and j in $\mathcal{J}$, an applicant a in $\mathcal{A}$ is said to *prefer* i to j if $r(a, i) < r(a, j)$; similarly, the applicant is *indifferent* between i and j if $r(a, i) = r(a, j)$.

For a given applicant a in $\mathcal{A}$, a's preference over jobs extends in a straightforward manner to matchings: Given matchings M and T, we say that a prefers matching M to T if a prefers $M(a)$ to $T(a)$, or if a is matched in M but not in T. Let $\phi(M, T)$ be the total number of applicants that prefer M to T:

$$\phi(M, T) = |\{a \in \mathcal{A} : a \text{ prefers } M \text{ to } T\}| . \tag{1}$$

We say that M is *more popular* than T and write $M \succ T$, if $\phi(M, T) > \phi(T, M)$. The matching M is *popular* if there is no matching more popular than M.

Definition 1. A matching M is popular if $\phi(M, T) \geq \phi(T, M)$ for every matching T.

When dealing with a set of independent agents, it is always desirable that the solution concept used is stable. Popular matchings have this property in the sense that an applicant majority vote cannot force a migration to another matching.

☆ The second author was supported by an Alexander von Humboldt Fellowship. Part of this work was done when the first author visited MPI für Informatik through the DST-MPG partner group on Efficient Graph Algorithms, IISc Bangalore.

* Corresponding author. Tel.: +49 681 9325 109.
E-mail addresses: kavitha@csa.iisc.ernet.in (T. Kavitha), jmestre@mpi-inf.mpg.de (J. Mestre), meghana@csa.iisc.ernet.in (M. Nasre).

1 Present address: Tata Institute of Fundamental Research, Mumbai, India.

0304-3975/$ – see front matter
doi:10.1016/j.tcs.2010.03.028

Thus a popular matching seems a stable and desirable answer to the question of how to assign applicants to jobs bearing their preferences in mind. However, popular matchings do not provide a complete answer since there are instances that do not admit any popular matching [9]. Consider the instance $\mathcal{A} = \{a_1, a_2, a_3\}$, $\mathcal{J} = \{j_1, j_2, j_3\}$ where all applicants rank the jobs in the same way, say j_1 is better than j_2, which in turn is better than j_3.

a_1	j_1 j_2 j_3
a_2	j_1 j_2 j_3
a_3	j_1 j_2 j_3

The following three matchings $M_1 = \{(a_1, j_1), (a_2, j_2), (a_3, j_3)\}$, $M_2 = \{(a_2, j_1), (a_3, j_2), (a_1, j_3)\}$, and $M_3 = \{(a_3, j_1), (a_1, j_2), (a_2, j_3)\}$ demonstrate that the more-popular-than relation need not be acyclic since $M_1 \prec M_2 \prec M_3 \prec M_1$. In fact, it is easy to see that this instance admits no popular matching. There are, however, efficient algorithms for determining if a given instance admits a popular matching and computing such a matching provided one exists [3].

In an attempt to deal with the issue that not every instance admits a popular matching, McCutchen [17] proposed two measures that captures how unpopular a matching is. Let $\Delta(M, T)$ be the difference between the number of applicants who prefer T and the number of applicants who prefer M, that is,

$$\Delta(M, T) = \phi(T, M) - \phi(M, T). \tag{2}$$

Let $\delta(M, T)$ be the ratio of number of applicants who prefer T to the number of applicants who prefer M, that is,

$$\delta(M, T) = \frac{\phi(T, M)}{\phi(M, T)}. \tag{3}$$

The *unpopularity margin* of M is defined as $\max_T \Delta(M, T)$ and the *unpopularity factor* of M is defined as $\max_T \delta(M, T)$. McCutchen showed that computing a matching M minimizing the unpopularity margin or unpopularity factor is NP-hard.

Hence, we are faced with the unpleasant prospect of choosing between a solution concept that can be computed efficiently but may not exist, or one that always exists but cannot be computed efficiently. The main contribution of our paper is to introduce a new solution concept based on the popularity criterion that has the best characteristics of previous work. Namely, it is guaranteed to exist and can be computed efficiently.

Mixed matchings. Motivated by the notions of pure and mixed strategies from Game Theory we propose to study popular mixed matchings. A *mixed matching* P is a set $\{(M_1, p_1), \ldots, (M_k, p_k)\}$, where $\sum_{i=1}^{k} p_i = 1$ and for each $i = 1, \ldots, k$, M_i is a matching in G and $p_i \geq 0$. Thus a mixed matching is simply a probability distribution over matchings in G and a pure matching M can be thought of as the mixed matching $\{(M, 1)\}$.

The function $\phi(M, T)$ that allowed us to compare two pure matchings M and T generalizes to mixed matchings in a natural way. For mixed matchings $P = \{(M_1, p_1), \ldots, (M_k, p_k)\}$ and $Q = \{(T_1, q_1), \ldots, (T_l, q_l)\}$ we let $\phi(P, Q)$ be the expected number of applicants that prefer M to T where M and T are drawn from the probability distributions P and Q respectively; in other words,

$$\phi(P, Q) = \sum_{i=1}^{k} \sum_{j=1}^{l} p_i \, q_j \, \phi(M_i, T_j). \tag{4}$$

We are now ready to give the definition of *popular mixed matching*.

Definition 2. A mixed matching P is popular if $\phi(P, Q) \geq \phi(Q, P)$ for all mixed matchings Q.

For example, consider the instance above on jobs $\{j_1, j_2, j_3\}$ and applicants $\{a_1, a_2, a_3\}$ with identical preference lists that admits no popular (pure) matching. Consider the mixed matching

$$P = \{(M_1, 1/3), (M_2, 1/3), (M_3, 1/3)\}.$$

It is easy to see that we have $\phi(P, T) \geq \phi(T, P)$ for all matchings T in this graph, which in turn implies that $\phi(P, Q) \geq \phi(Q, P)$ for all mixed matchings Q in this graph. Thus the mixed matching P is popular in the instance above.

The rest of the paper is organized as follows. In Section 2 we prove that every instance admits a popular mixed matching by establishing a connection with a certain exponentially-large zero-sum game. In Section 3 we give efficient algorithms for solving this large game, which translate into efficient algorithms for computing a popular mixed matching. Our technique applies to a larger class of games and may be of independent interest. In Section 4, we give tight bounds on the efficiency of mixed popular matchings using the standard measures of price of anarchy and price of stability. Finally, in Section 5 we show that our approach can be modified to handle several generalizations of the basic popular matching problem.

1.1. Related work

Popular matchings were first studied by Gardenfors [9] in the context of the stable marriage problem where each side has preferences over members of the other side. When only one side has preferences, Abraham et al. [3] gave polynomial time algorithms to find a popular matching, or to report none exists. Their algorithm takes $O(m + n)$ time when the preference lists are strictly ordered and when the preference lists contain ties, they gave an $O(m\sqrt{n})$ time algorithm, where m is in the number of edges and n is the number of vertices. Subsequently, Mahdian [13] showed that a popular matching exists with high probability, when preference lists are randomly constructed, and the number of jobs is a factor of $\alpha \approx 1.42$ larger than the number of applicants. In fact, he showed that a phase transition occurs at α; namely, if the ratio $\frac{|\mathcal{J}|}{|\mathcal{A}|}$ is smaller than α then with high probability popular matchings do not exist.

Manlove and Sng [16] generalized the algorithms of [3] to the case where each job has an associated *capacity*, the number of applicants that it can accommodate. Mestre [18] designed an efficient algorithm for the *weighted* popular matching problem, where each applicant is assigned a priority or weight and the definition of popularity takes into account these priorities.

McCutchen [17] proposed two quantities to measure the *unpopularity* of a matching and showed polynomial time algorithms to compute these measures for any fixed matching. He also showed that the problem of computing a matching that minimizes either of these measures is NP-hard. Huang et al. [11] gave algorithms to compute matchings with bounded values of these unpopularity measures in certain graphs.

The topic of mixed matchings has been studied extensively in the Economics literature; we are aware of [12,5,4,21, 2,1,14,15]. This line of research is concerned with designing mechanisms that have a number of properties, such as envy-free, Pareto optimal, and strategy-proof. These properties can be defined in slightly different ways—the interested reader is referred to the article by Katta et al. [12] for an excellent overview of the various definitions commonly used. Roughly speaking, a mixed matching is envy-free if no applicant prefers to get other applicant's allocation to his own; it is Pareto optimal if it is not possible to improve someone's allocation without hurting someone else; and the mechanism is strategy proof if the applicants do not have an incentive to lie about their true preferences. Depending on how one defines an applicant's preference over mixed matchings it may [12,5] or may not [21] be possible to achieve simultaneously all these properties. Note that popular matchings take an orthogonal approach to this, placing emphasis not on individual applicants but on their aggregate or majority.

2. Existence of popular mixed matchings

Recall that our input is a bipartite graph $G = (\mathcal{A} \cup \mathcal{J}, E)$ with $n = |\mathcal{A}| + |\mathcal{J}|$ vertices and $m = |E|$ edges, where each applicant $a \in \mathcal{A}$ has a preference list (can include ties) over its neighboring jobs. For ease of exposition we will introduce a unique last-resort job ℓ_a for every $a \in \mathcal{A}$, which we append at the end of a's preference list. This modification does not change the fact of whether the instance has a popular (mixed) matching or not, but it has the benefit that we can restrict our attention to applicant-complete assignments.

The instance on 3 applicants and 3 jobs described in the previous section had a popular mixed matching P that assigns each job j_i to each of the applicants a_1, a_2, a_3 with a probability of 1/3. This might lead one to conclude that partitioning a job "equally" among applicants who rank this job as a top job (similarly, as a second job) yields a popular mixed matching. The following instance shows that this is however not the case. Let $\mathcal{A} = \{a_1, a_2, b_1, b_2, x\}$ and $\mathcal{J} = \{j_1, j_2, i_1, i_2\}$, along with the last resort jobs. The following table describes the preference lists of applicants.

a_1	j_1	j_2	ℓ_{a_1}
a_2	j_1	j_2	ℓ_{a_2}
b_1	i_1	i_2	ℓ_{b_1}
b_2	i_1	i_2	ℓ_{b_2}
x	j_1	i_2	ℓ_x

Applicants a_1 and a_2 have identical preference lists: their first choice is j_1 and their second choice is j_2. Applicants b_1 and b_2 have identical preference lists: their first choice is i_1 and their second choice is i_2. Applicant x has its first choice as j_1 and second choice as i_2. Using the characterization given in [3] for when a given instance admits popular (pure) matchings,[2] it is easy to see that this instance admits no popular pure matching.

An *equal* partition would divide j_1 with a share of 1/3 to each of a_1, a_2 and x, and divide i_1 with a share of 1/2 to each of b_1 and b_2. Then, it would proceed to divide j_2 with a share of 1/2 to a_1 and a_2, and divide i_2 with a share of 1/3 to each of b_1, b_2 and x. Finally, a_1, a_2, b_1, b_2 get 1/6th share of their respective last resort jobs while x gets a 1/3rd share of ℓ_x. As we will see shortly in Section 2, we can construct an equivalent allocation as a mixed matching $\{(M_i, p_i) : 1 \le i \le k\}$. This allocation, however, is *not* popular as the matching $M_0 = \{(a_1, j_1), (a_2, j_2), (b_1, i_1), (x, i_2), (b_2, \ell_{b_2})\}$ is more popular than it. The justification of this claim involves some tedious calculations, so it appears in Appendix.

On the other hand consider the matching $P = \{(M_0, 1/2), (M_1, 1/2)\}$ where M_0 is the above matching and $M_1 = \{(a_1, j_2), (a_2, j_1), (x, i_2), (b_1, \ell_{b_1}), (b_2, i_1)\}$. As we shall prove in Section 3, P is indeed popular and therefore the instance admits a popular mixed matching.

Theorem 1. *Every instance admits a popular mixed matching.*

[2] For strict preference lists, there has to be an $\mathcal{A}$-perfect matching in the subgraph where each applicant is adjacent to its top choice job and its most preferred job that is not top choice for any applicant.

Proof. We will model our problem as a two-person zero-sum game. The rows and columns of the payoff matrix S are indexed by all the possible matchings $M_1, \ldots, M_N$ in G. The (i,j)-th entry of the matrix S is $\Delta(M_i, M_j)$. Recall that $\Delta(M_i, M_j)$ is the difference between the number of applicants who prefer M_j to M_i and the number of applicants who prefer M_i to M_j. A mixed strategy of the row player is a probability distribution $\pi = \langle p_1, \ldots, p_N \rangle$ over the rows of S; similarly, a mixed strategy of the column player is a probability distribution $\sigma = \langle q_1, \ldots, q_N \rangle$ over the columns of S.

The row player seeks a strategy π so that $\max_\sigma \Delta(\pi, \sigma)$ is minimized, where $\Delta(\pi, \sigma) = \sum_{i=1}^N \sum_j^N p_i\, q_j\, \Delta(M_i, M_j)$. Notice that π and σ can also be regarded as mixed matchings, in which case we can think of the row player as trying to find a mixed matching with least *expected* unpopularity margin.

It follows that the given instance admits a popular mixed matching if and only if

$$\min_\pi \max_\sigma \sum_{i=1}^N \sum_{j=1}^N p_i\, q_j \Delta(M_i, M_j) \le 0.$$

It is easy to see that

$$\min_\pi \max_\sigma \sum_{i=1}^N \sum_{j=1}^N p_i\, q_j \Delta(M_i, M_j) \ge \min_\pi \sum_{i,j=1}^N p_i\, p_j\, \Delta(M_i, M_j) = 0 \tag{5}$$

where the inequality follows from taking $\sigma = \pi$ and the equality from the fact that $\Delta(M_i, M_j) = -\Delta(M_j, M_i)$ for all i and j. Thus $\min_\pi \max_\sigma \Delta(\pi, \sigma)$ must be 0 if the instance admits a popular mixed matching.

The column player seeks a strategy σ so that $\min_\pi \Delta(\pi, \sigma)$ is maximized. By von Neumann's Minimax Theorem [19] we have

$$\min_\pi \max_\sigma \sum_{i,j=1}^N p_i\, q_j\, \Delta(M_i, M_j) = \max_\sigma \min_\pi \sum_{i,j=1}^N p_i\, q_j\, \Delta(M_i, M_j). \tag{6}$$

We can bound the right-hand side of (6) analogous to (5) to get

$$\max_\sigma \min_\pi \sum_{i,j=1}^N p_i\, q_j\, \Delta(M_i, M_j) \le \max_\sigma \sum_{i,j=1}^N q_i\, q_j\, \Delta(M_i, M_j) = 0. \tag{7}$$

Combining Eqs. (5), (6) and (7) we get $\min_\pi \max_\sigma \Delta(\pi, \sigma) = 0$. Therefore every instance admits a popular mixed matching. □

3. Finding a popular mixed matching

The proof of Theorem 1 implicitly provides an algorithm for computing a popular mixed matching. Namely, given a zero-sum game, its value and mixed strategies attaining that value can be computed using linear programming. Here we need to determine $p_1, \ldots, p_N$ adding up to 1, such that $\sum_{i=1}^N p_i\, \Delta(M_i, T) \le 0$ for all pure matchings T and $p_i \ge 0$ for all i.

Unfortunately, the linear program (shown below) is too large to be solved efficiently.

$$\text{minimize } \tau \tag{LP1}$$

subject to

$$\begin{aligned} \textstyle\sum_i p_i\, \Delta(M_i, T) &\le \tau \quad \forall \text{ matchings } T \\ \textstyle\sum_i p_i &= 1 \\ p_i &\ge 0 \quad \forall i = 1, \ldots, N. \end{aligned}$$

Because the number of variables *and* the number of constraints in (LP1) is typically exponential, the ellipsoid algorithm cannot be applied directly. In order to do so, we must first reduce the number of variables. This can be achieved by working with *fractional matchings* instead of mixed matchings.

Let $\mathcal{X}$ be the set of fractional $\mathcal{A}$-complete matchings in G, that is,

$$\mathcal{X} = \left\{ \vec{x} \in \mathbb{R}_+^m \;\middle|\; \begin{array}{l} \sum_{e \in E(a)} \vec{x}(e) = 1 \text{ for } a \in \mathcal{A} \text{ and} \\ \sum_{e \in E(j)} \vec{x}(e) \le 1 \text{ for } j \in \mathcal{J}. \end{array} \right\} \tag{8}$$

where $E(u)$ is the set of edges incident on vertex u.

Let $\widehat{\mathcal{X}}$ be the set of extreme points of $\mathcal{X}$. Note that because $\mathcal{X}$ is integral we have $\widehat{\mathcal{X}} = \mathcal{X} \cap \{0, 1\}^m$, which corresponds to the set of integral $\mathcal{A}$-complete matchings in G.

Clearly, every mixed matching $P = \langle p_1, \dots, p_N \rangle$ induces a fractional matching $\vec{x} = \sum_{M_i} p_i I_{M_i}$, where $I_{M_i} \in \{0, 1\}^m$ is the characteristic vector of the matching M_i. In turn, every $\vec{x} \in \mathcal{X}$ can be expressed as a convex combination of the extreme points of $\mathcal{X}$. This implies a many-to-one mapping between mixed and fractional matchings. Carathéodory's Theorem [6] states that if a vector $\vec{x}$ in R^m lies in the convex hull of the set $\widehat{\mathcal{X}}$, then $\vec{x}$ can be written as the convex combination of at most $m + 1$ points in $\widehat{\mathcal{X}}$. Therefore, given a fractional matching, we can always get an equivalent mixed matching whose support is no larger than $m + 1$.

The plan, therefore, is to replace the mixed matching in (LP1) with a fractional matching. In order to do so, we need to define the function $\Delta(\cdot)$ for fractional matchings $\vec{x_1}, \vec{x_2} \in \mathcal{X}$:

$$\Delta(\vec{x_1}, \vec{x_2}) = \sum_{a \in \mathcal{A}} \sum_{\substack{(a,i) \in E(a) \\ (a,j) \in E(a)}} \vec{x_1}(a, i)\, \vec{x_2}(a, j)\, \mathsf{vote}_a(i, j) \tag{9}$$

where the term $\mathsf{vote}_a(i, j)$ captures a's willingness to switch from i to j:

$$\mathsf{vote}_a(i, j) = \begin{cases} -1 & \text{if } a \text{ prefers } i \text{ over } j, \\ 1 & \text{if } a \text{ prefers } j \text{ over } i, \\ 0 & \text{if } a \text{ is indifferent between } i \text{ and } j. \end{cases} \tag{10}$$

Lemma 1. *Let P and Q be two mixed matchings and let $\vec{x_1}$ and $\vec{x_2}$ be their corresponding fractional matchings. Then $\Delta(P, Q) = \Delta(\vec{x_1}, \vec{x_2})$.*

Proof. P and Q are probability distributions $\langle p_1, \dots, p_N \rangle$ and $\langle q_1, \dots, q_N \rangle$ respectively, over matchings $M_1, \dots, M_N$.

$$\begin{aligned} \Delta(P, Q) &= \sum_{g,h=1}^{N} p_g\, q_h\, \Delta(M_g,\ M_h) \\ &= \sum_{g,h=1}^{N} p_g\, q_h \sum_{a \in \mathcal{A}} \mathsf{vote}_a(M_g(a), M_h(a)) \\ &= \sum_{g,h=1}^{N} p_g\, q_h \sum_{a \in \mathcal{A}} \sum_{\substack{(a,i) \in E(a) \\ (a,j) \in E(a)}} I_{M_g}(a, i)\, I_{M_h}(a, j)\, \mathsf{vote}_a(i, j) \\ &= \sum_{a \in \mathcal{A}} \sum_{\substack{(a,i) \in E(a) \\ (a,j) \in E(a)}} \sum_{g,h=1}^{N} p_g\, q_h\, I_{M_g}(a, i)\, I_{M_h}(a, j)\, \mathsf{vote}_a(i, j). \end{aligned}$$

Regrouping the terms in the innermost sum we get

$$\begin{aligned} \Delta(P, Q) &= \sum_{a \in \mathcal{A}} \sum_{\substack{(a,i) \in E(a) \\ (a,j) \in E(a)}} \left(\sum_{g=1}^{N} p_g\, I_{M_g}(a, i) \right) \left(\sum_{h=1}^{N} q_h\, I_{M_h}(a, j) \right) \mathsf{vote}_a(i, j) \\ &= \sum_{a \in \mathcal{A}} \sum_{\substack{(a,i) \in E(a) \\ (a,j) \in E(a)}} \vec{x_1}(a, i)\, \vec{x_2}(a, j)\, \mathsf{vote}_a(i, j) \\ &= \Delta(\vec{x_1}, \vec{x_2}). \quad \square \end{aligned}$$

It should be clear now that the following linear program is equivalent to (LP1).

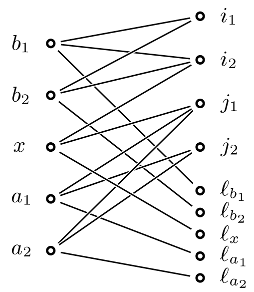
$$\text{minimize } \tau \tag{LP2}$$

subject to

$$\begin{aligned} \Delta(\vec{x}, T) \le \tau \quad & \forall \text{ matchings } T \\ \vec{x} \in \mathcal{X} & \end{aligned}$$

Unlike (LP1), the linear program (LP2) has only $m+1$ variables: τ and the coordinates of $\vec{x}$. This new program can be solved in polynomial time using the ellipsoid method. To prove this we only need to design a polynomial-time separation oracle, which given an infeasible solution $(\vec{x}, \tau)$ returns a violated constraint of (LP2). It is trivial how to test $\vec{x} \in \mathcal{X}$ efficiently; so we only need to test whether there is a matching T such that $\Delta(\vec{x}, T) > \tau$. This is done by computing the *unpopularity margin* of $\vec{x}$.

Definition 3. The unpopularity margin of $\vec{x} \in \mathcal{X}$ is $\max_T \Delta(\vec{x}, T)$.



$$
\begin{array}{lll}
w_{\vec{x}}(a_k, j_1) = 0.5 & w_{\vec{x}}(a_k, j_2) = -0.5 & w_{\vec{x}}(a_k, \ell_{a_k}) = -1 \\
w_{\vec{x}}(b_k, i_1) = 0.5 & w_{\vec{x}}(b_k, i_2) = 0 & w_{\vec{x}}(b_k, \ell_{b_k}) = -0.5 \\
w_{\vec{x}}(x, j_1) = 1 & w_{\vec{x}}(x, i_2) = 0 & w_{\vec{x}}(x, \ell_x) = -1
\end{array}
$$

Fig. 1. Computing the unpopularity margin of the fractional matching $\vec{x}(a_k, j_1) = \vec{x}(a_k, j_2) = 0.5$ for $k = 1, 2$; $\vec{x}(x, i_2) = 1$; and $\vec{x}(b_k, i_1) = \vec{x}(b_k, \ell_{b_k}) = 0.5$ for $k = 1, 2$. The bipartite graph G is shown on the left and weights $w_{\vec{x}}$ are given on the right.

As mentioned earlier, McCutchen [17] studied the above measure for pure matchings. He gave a polynomial time algorithm based on min-cost flows to measure the unpopularity margin $\max_T \Delta(M, T)$ of a given matching M. We describe his algorithm below in the equivalent language of the maximum weight assignment problem and then show that it can be used to compute the unpopularity margin of a fractional matching also.

McCutchen's algorithm to determine the unpopularity margin of a matching T. Given a matching M we want to compute its unpopularity margin. The idea is to define a weight function w_M on the edges of G such that for any matching T, the weight of T under w_M equals $\Delta(M, T)$. Computing the unpopularity margin of M then reduces to solving the maximum weight assignment problem (G, w_M), which can be done in $O(n\,(m+n\log n))$ time [8].

The quantity $\Delta(M, T)$ can be expressed as a sum of individual votes $\sum_{a\in\mathcal{A}} \mathsf{vote}_a(T(a), M(a))$. Therefore, setting $w_M(a, j) = \mathsf{vote}_a(j, M(a))$ achieves the desired effect that $\Delta(M, T) = w_M(T)$.

McCutchen's method readily generalizes to fractional matchings. Let $\vec{x}$ be a fractional matching. We define the weight of an edge $(a, j) \in E$ as

$$w_{\vec{x}}(a, j) \;=\; \sum_{(a,i)\in E(a)} \vec{x}(a, i)\, \mathsf{vote}_a(i, j). \tag{11}$$

It is straightforward to verify that indeed $\Delta(\vec{x}, T) = w_{\vec{x}}(T)$. Computing the unpopularity margin of a fractional matching then reduces to computing a maximum weight assignment.

It is instructive to look at a concrete example to see the above reduction in action. Fig. 1 describes the reduction for the instance described at the beginning of Section 2 and the fractional matching corresponding to the mixed matching $P = \{(M_0, 1/2), (M_1, 1/2)\}$ where $M_0 = \{(a_1, j_1), (a_2, j_2), (x, i_2), (b_1, i_1), (b_2, \ell_{b_2})\}$ and $M_1 = \{(a_1, j_2), (a_2, j_1), (x, i_2), (b_1, \ell_{b_1}), (b_2, i_1)\}$. It is easy to see that a maximum weight applicant-complete matching in this graph has weight 0. This proves that the mixed matching P is indeed popular.

The procedure for computing the unpopularity margin of a fractional matching provides the necessary oracle to apply the ellipsoid method to solve (LP2). Therefore we can find a popular mixed matching in polynomial time. While this settles the complexity of our problem, the scheme presented is not truly efficient as the ellipsoid algorithm is notoriously slow in practice. We will address this issue now by giving an alternative linear programming formulation whose size is only linear in the size of G.

Theorem 2. *There exists a linear programming formulation for finding a popular fractional matching with $m + n$ variables and constraints.*

Proof. Let us start by rewriting (LP2) as a mathematical program

$$\min_{\vec{x}\in\mathcal{X}} \max_{\vec{z}\in\widehat{\mathcal{X}}} w_{\vec{x}}(\vec{z}). \tag{12}$$

Because $\mathcal{X}$ is integral, the mathematical program (12) is in fact equivalent to the following program

$$\min_{\vec{x}\in\mathcal{X}} \max_{\vec{y}\in\mathcal{X}} w_{\vec{x}}(\vec{y}). \tag{13}$$

The main obstacle that we must overcome in order to formulate (13) as a pure linear program is the non-linear objective $w_{\vec{x}}(\vec{y})$. We can overcome this hurdle with the aid of linear programming duality. Consider the primal program $\max\,\{w_{\vec{x}}(\vec{y}) : \vec{y} \in \mathcal{X}\}$ and let $\mathcal{D}_{\vec{x}}$ denote the feasible region of its dual:

$$\mathcal{D}_{\vec{x}} = \left\{ \vec{\alpha} \in \mathbb{R}^{|\mathcal{A}|},\ \vec{\beta} \in \mathbb{R}_+^{|\mathcal{J}|} : \vec{\alpha}(a) + \vec{\beta}(j) \geq w_{\vec{x}}(a, j) \text{ for each } (a, j) \in E \right\}.$$

By the Strong Duality Theorem we get

$$\max_{\vec{y}\in\mathcal{X}} w_{\vec{x}}(\vec{y}) \;=\; \min_{(\vec{\alpha},\vec{\beta})\in\mathcal{D}_{\vec{x}}} \sum_{a\in\mathcal{A}} \vec{\alpha}(a) + \sum_{j\in\mathcal{J}} \vec{\beta}(j).$$

Therefore, (13) is equivalent to the following succinct linear program

$$\text{minimize} \sum_{a\in\mathcal{A}} \vec{\alpha}(a) + \sum_{j\in\mathcal{J}} \vec{\beta}(j) \tag{LP3}$$

subject to

$$\begin{aligned}
&\vec{\alpha}(a) + \vec{\beta}(j) \geq \textstyle\sum_{(a,i)\in E(a)} \vec{x}(a,i)\,\text{vote}_a(i,j) && \forall\,(a,j)\in E\\
&\sum_{(a,j)\in E(a)} \vec{x}(a,j) = 1 && \forall\, a\in\mathcal{A}\\
&\sum_{(a,j)\in E(j)} \vec{x}(a,j) \leq 1 && \forall\, j\in\mathcal{J}\\
&\vec{x}\in\mathbb{R}_+^m\\
&\vec{\beta}\in\mathbb{R}_+^{|\mathcal{J}|}\\
&\vec{\alpha}\in\mathbb{R}^{|\mathcal{A}|}
\end{aligned}$$

Recall that (13) in turn is equivalent to (LP1). The new linear program has only $m+n$ variables and $m+n$ constraints. □

4. Efficiency of popular mixed matchings

Now that we have settled the existence and computability of popular mixed matchings, the next logical step is to study how efficient this solution concept is. Recall that we inserted a last resort job for each applicant. Perhaps the most natural measure of efficiency one can define is the number of applicants that are assigned a real job.

Let w be a weight function $E \to \{0, 1\}$ defined as

$$w(a,j) = \begin{cases} 0 & \text{if } j = \ell(a)\\ 1 & \text{otherwise}\end{cases}$$

then the expected size of a fractional matching $\vec{x}$ is defined as

$$w(\vec{x}) = \sum_{(a,j)\in E} w(a,j)\,\vec{x}(a,j).$$

Notice that we can optimize this quantity by using a modified version of (LP3): Simply add the constraint $\sum_{a\in\mathcal{A}} \vec{\alpha}(a) + \sum_{j\in\mathcal{J}} \vec{\beta}(j) = 0$ and change the objective to maximize $w(\vec{x})$.

Of course, when we restrict our attention to popular matchings we may be ignoring larger matchings. The question we would like to investigate is how do popular mixed matchings compare to a maximum size matching that is not restricted by the popularity requirement. The standard analytical tool to do so is to look at the price of anarchy and the price of stability of the popular matching problem.

Recall that $\mathcal{X}$ is the set of fractional matchings (8). Let $\mathcal{R}$ be the subset of fractional matchings that are popular. The *price of anarchy* of the popular matching problem is defined as

$$\sup \frac{\max_{\vec{x}\in\mathcal{X}} w(\vec{x})}{\min_{\vec{r}\in\mathcal{R}} w(\vec{r})}, \tag{14}$$

and its *price of stability* is defined as

$$\sup \frac{\max_{\vec{x}\in\mathcal{X}} w(\vec{x})}{\max_{\vec{r}\in\mathcal{R}} w(\vec{r})}, \tag{15}$$

where the suprema range over all possible instances of the popular matching problem.

The price of anarchy bounds the efficiency of any popular mixed matching compared to the best possible (not necessarily popular) matching, while the price of stability captures the efficiency of the best popular mixed matching in the worst case.

Theorem 3. *The price of anarchy and the price of stability of the popular matching problem is* 2.

Proof. The proof has two parts. First we show that the price of anarchy is at most 2. Then we show that the price of stability is at least 2. The theorem follows from these two facts, since the price of stability cannot be larger than the price of anarchy.

Let T be pure matching in $\mathcal{X}$ and let $\vec{s}$ be any popular matching in $\mathcal{R}$. Since $\vec{s}$ is popular, we have

$$\mathrm{E}_{M\leftarrow\vec{s}}[\phi(M,T)] \geq \mathrm{E}_{M\leftarrow\vec{s}}[\phi(T,M)].$$

Now bounding on both sides using the fact that $\phi(M,T) \leq w(M)$ and $\phi(T,M) \geq w(T) - w(M)$, we obtain

$$\mathrm{E}_{M\leftarrow\vec{s}}[w(M)] \geq \mathrm{E}_{M\leftarrow\vec{s}}[w(T) - w(M)].$$

And thus $2w(s) \geq w(T)$ and the price of anarchy is at most 2.

For bounding the price of stability, consider the following instance with $2k$ applicants $\{a_1, \ldots, a_{2k}\}$ and $2k+1$ jobs $\{j_0, \ldots, j_{2k}\}$ (we do not show here the last resort jobs).

a_1	j_0	j_1	a_{k+1}	j_1	j_{k+1}
a_2	j_0	j_2	a_{k+2}	j_2	j_{k+2}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
a_k	j_0	j_k	a_{2k}	j_k	j_{2k}

The matching $\{(a_t, j_t) : 1 \leq t \leq 2k\}$ is maximum and has weight $2k$. We claim that every popular matching must assign p_t in full to a_{k+t}. This implies that every popular matching has weight at most $k+1$; thus, as k grows, the price of stability for the instance tends to 2. Given a popular mixed matching $\vec{s}$, we construct a new fractional matching as follows

$$\vec{r}(a_t, j) = \begin{cases} \dfrac{1}{k} & \text{if } 1 \leq t \leq k \text{ and } j = j_0 \\ \dfrac{k-1}{k} & \text{if } 1 \leq t \leq k \text{ and } j = \ell_{a_t} \\ 1 & \text{if } k < t \leq 2k \text{ and } j = j_t. \end{cases}$$

Since $\vec{s}$ is popular, we must have $\Delta(\vec{r}, \vec{s}) \geq 0$. Expanding the left-hand side of this inequality we get

$$\begin{aligned} \Delta(\vec{r},\vec{s}) &= \sum_{t=1}^{k} \Big[\vec{r}(a_t, \ell_{a_t})\big(1 - \vec{s}(a_t, \ell_{a_t})\big) - \vec{r}(a_t, j_0)\big(1 - \vec{s}(a_t, j_0)\big) \Big] + \sum_{t=1}^{k} \Big[0 - (1 - \vec{s}(a_{k+t}, j_t)) \Big], \\ &= \sum_{t=1}^{k} \left[\frac{k-1}{k} \big(\vec{s}(a_t, j_0) + \vec{s}(a_t, j_t)\big) - \frac{1}{k}\big(1 - \vec{s}(a_t, j_0)\big) \right] - \sum_{t=1}^{k} \big(1 - \vec{s}(a_{k+t}, j_t)\big). \end{aligned}$$

Without loss of generality, we can assume that $\sum_t \vec{s}(a_t, j_0) = 1$ and $\vec{s}(a_{k+t}, j_t) = 1 - \vec{s}(a_t, j_t)$, for otherwise $\vec{s}$ would not be popular. Using these two observations, the above equality simplifies to

$$\Delta(\vec{r},\vec{s}) = \sum_{t=1}^{k} -\frac{\vec{s}(a_t, j_t)}{k}.$$

Therefore, since $\Delta(\vec{r},\vec{s}) \geq 0$, we have that $\vec{s}(a_t, j_t) = 0$ and $\vec{s}(a_{k+t}, j_t) = 1$ for all $1 \leq t \leq k$. □

5. Generalizations

So far we have focused on the simplest setting where all applicants are treated equally, each job can accept a single applicant, and preferences are one-sided. In this section, we show how to extend our results to the setting where applicants have priorities or weights [18], where jobs have capacities [16], or where both jobs and applicants have preferences [9].

In order to obtain these results we first prove a theorem about a certain class of games called *polyhedral zero-sum games*. Then for each variant of the popular matching problem we re-define the function ϕ that compares matchings and the polytope $\mathcal{X}$ of fractional matchings to cast the particular variant at hand as a polyhedral zero-sum game.

5.1. Polyhedral zero-sum games

In this section we show that our results from Section 3 extend to a larger class of zero-sum games that we call *polyhedral zero-sum games*. Consider a two player zero-sum game in which the strategies of the players are the extreme points of some polytope. Specifically, let $\mathcal{P}$ be a polytope in R^d defined by f constraints. Let us denote by $\widehat{\mathcal{P}}$ the set of extreme points of $\mathcal{P}$. The payoff function for two $\vec{x}, \vec{y} \in \widehat{\mathcal{P}}$ is a bilinear function

$$\text{payoff}(\vec{x}, \vec{y}) = \sum_{i,j=1}^{d} w_{ij} \vec{x}(i) \vec{y}(j),$$

where the w_{ij} coefficients are part of the game definition. A mixed strategy for the game is convex combination of extreme points $\sum_{i=1}^{N} p_i \vec{x}_i$, where $\vec{x}_i \in \widehat{\mathcal{P}}$, $p_i \geq 0$ for all i, and $\sum_{i=1}^{N} p_i = 1$; notice that the resulting convex combination is just a point in $\mathcal{P}$. Our goal is to compute an optimal mixed strategy for such a game. This can easily be done by a linear program with d variables and $|\widehat{P}|$ constraints. However, this is not very efficient as $|\widehat{P}|$ can be exponentially large compared to $d + f$. The following theorem provides a more compact formulation for computing an optimal mixed strategy.

Theorem 4. *There exists a linear programming formulation with $d + f$ variables and constraints for finding an optimal mixed strategy for a general polyhedral zero-sum game.*

Proof. Suppose $\mathcal{P}$ is defined by the following f constraints:

$$\mathcal{P} = \left\{ \vec{x} \in \mathbb{R}_+^d : \sum_{j=1}^{d} A_{ij} \vec{x}(j) \leq b_i \quad \forall i = 1 \cdots f \right\}. \tag{16}$$

Our goal is to compute an optimal mixed strategy

$$\min_{\vec{x} \in \mathcal{P}} \max_{\vec{y} \in \mathcal{P}} \sum_{i,j=1}^{d} w_{ij} \vec{x}(i) \vec{y}(j). \tag{17}$$

As before, we have a non-linear objective function. In order to overcome this we again resort to linear programming duality. For a fixed $\vec{x}$ consider the linear program

$$\max_{\vec{y} \in \mathcal{P}} \sum_{i,j=1}^{d} w_{ij} \vec{x}(i) \vec{y}(j).$$

Its dual is given by

$$\min_{\beta \in \mathcal{D}_{\vec{x}}} \sum_{i=1}^{f} b_i \vec{\beta}(i),$$

where $\mathcal{D}_{\vec{x}}$ is the feasible dual region

$$\mathcal{D}_{\vec{x}} = \left\{ \vec{\beta} \in \mathbb{R}_+^f : \sum_{i=1}^{f} \vec{\beta}(i) A_{ij} \geq \sum_{i=1}^{d} w_{ij} \vec{x}(i) \;\; \forall j = 1 \cdots d \right\}.$$

Using the Strong Duality Theorem, we can now re-write (17) as a pure linear program.

$$\text{minimize} \sum_{i=1}^{f} b_i \vec{\beta}(i) \tag{LP4}$$

subject to

$$\sum_{i=1}^{f} A_{ij} \vec{\beta}(i) \geq \sum_{i=1}^{d} w_{ij} \vec{x}(i) \qquad \forall j = 1 \cdots d$$

$$\sum_{j=1}^{d} A_{ij} \vec{x}(j) \leq b_i \qquad \forall i = 1 \cdots f$$

$$\vec{x} \in \mathbb{R}_+^d$$

$$\vec{\beta} \in \mathbb{R}_+^f$$

The above linear program has $d + f$ variables and constraints. Let $(\vec{x}, \vec{\beta})$ be an optimal solution to (LP4). Then $\sum_{i=1}^{f} b_i \vec{\beta}(i)$ is the value of the polyhedral game and $x \in \mathcal{P}$ is an optimal mixed strategy. □

We now show how to apply Theorem 4 to the different variants of the popular matching problem.

5.2. Weighted popular mixed matchings

In this case, every applicant $a \in \mathcal{A}$ is assigned a positive weight $t(a)$. As mentioned earlier, the weighted case was first considered by Mestre [18] who gave efficient algorithms to determine if a given instance admits a popular weighted matching and to compute one if it exists.

Let $G(\mathcal{A} \cup \mathcal{J}, E)$ be an instance of the weighted popular matchings problem where every applicant $a \in \mathcal{A}$ has a weight $t(a)$ associated with it. For any two pure matchings M and T in G the function $\phi(M, T)$ is defined as follows:

$$\phi(M, T) = \sum_{\substack{a \in \mathcal{A}: \\ a \text{ prefers } M \text{ to } T}} t(a). \tag{18}$$

The definition of mixed popular matchings is similar to the unweighted case using (18) instead of (1). The set $\mathcal{X}$ of fractional matchings in the weighted case remains unchanged and is given by (8). Recall that $\mathcal{X}$ is integral. We extend the operator Δ for comparing mixed matchings to fractional matchings $\vec{x_1}, \vec{x_2} \in \mathcal{X}$.

$$\Delta(\vec{x_1}, \vec{x_2}) = \sum_{a \in \mathcal{A}} \sum_{\substack{(a,i) \in E(a) \\ (a,j) \in E(a)}} \vec{x_1}(a, i)\, \vec{x_2}(a, j)\, \mathsf{vote}_a(i, j), \tag{19}$$

where

$$\mathsf{vote}_a(i, j) = \begin{cases} -t(a) & \text{if } a \text{ prefers } i \text{ over } j, \\ t(a) & \text{if } a \text{ prefers } j \text{ over } i, \\ 0 & \text{if } a \text{ is indifferent between } i \text{ and } j. \end{cases}$$

Lemma 2. *Let P and Q be two mixed matchings in a weighted popular matchings instance G and let $\vec{x_1}$ and $\vec{x_2}$ be their corresponding fractional matchings. Then $\Delta(P, Q) = \Delta(\vec{x_1}, \vec{x_2})$.*

We omit the proof of Lemma 2 as it is identical to the proof of Lemma 1.

Theorem 5. *A weighted popular mixed matching always exists and can be computed efficiently.*

Proof. We define the polyhedral game for the polytope $\mathcal{X}$ using the payoff matrix Δ. Because the payoff matrix is skewed symmetric the value of the game is zero and thus there always exists a weighted popular mixed matching.

To compute such a mixed matching we note that the payoff matrix of the game defined by (19) is bilinear. Thus we can apply Theorem 4 to obtain a compact linear programming formulation for finding a weighted popular mixed matching. □

5.3. Capacitated popular mixed matchings

In this subsection we consider the generalization where every job $j \in \mathcal{J}$ has a positive capacity c_j associated with it. A solution to the capacitated matchings problem allows every applicant to be matched to at most one job and every job to be matched to at most c_j applicants. Given an instance of the capacitated popular matchings problem, we can make c_j "clones" of job j and create a new instance where every job has unit capacity. Our results for the uncapacitated case immediately imply the existence and computation of popular mixed matching for the capacitated case. Here instead of using this reduction we work directly with the capacitated instance. As a result we obtain more efficient algorithms.

Let $G(\mathcal{A} \cup \mathcal{J}, E)$ be an instance of the capacitated popular matchings problem where each job $j \in \mathcal{J}$ has a positive capacity c_j associated with it. For two (capacitated) matchings M and T in G, the function $\phi(M, T)$ is as defined for the uncapacitated case. That is,

$$\phi(M, T) = |\{a \in \mathcal{A} : a \text{ prefers } M \text{ over } T\}| .$$

The definition mixed popular matchings remains the same as in the uncapacitated case. The set of fractional matchings $\mathcal{X}$, however, changes since every job can now be matched to up to c_j applicants.

$$\mathcal{X} = \left\{ \vec{x} \in \mathbb{R}_+^m \;\middle|\; \begin{array}{l} \sum_{e \in E(a)} \vec{x}(e) = 1 \text{ for } a \in \mathcal{A} \text{ and} \\ \sum_{e \in E(j)} \vec{x}(e) \le c_j \text{ for } j \in \mathcal{J}. \end{array} \right\} \tag{20}$$

Note that $\mathcal{X}$ is still integral as the LP matrix is totally unimodular. The definition of Δ to compare fractional matchings is the same as (9). The following lemma allows us to replace mixed matchings with fractional matchings in the capacitated case.

Lemma 3. *Let P and Q be two mixed matchings in a capacitated popular matchings instance G and let $\vec{x_1}$ and $\vec{x_2}$ be their corresponding fractional matchings. Then $\Delta(P, Q) = \Delta(\vec{x_1}, \vec{x_2})$.*

We omit the proof of Lemma 3 as it is identical to the proof of Lemma 1.

Theorem 6. *A capacitated popular mixed matching always exists and can be computed efficiently.*

Proof. We define the polyhedral game for the polytope $\mathcal{X}$ defined by (20) using the payoff matrix Δ. Because the payoff matrix is skewed symmetric the value of the game is zero and thus there always exists a capacitated popular mixed matching.

To compute such a mixed matching we note that the payoff matrix of the game defined by (9) is bilinear. Thus we can apply Theorem 4 to obtain a compact linear programming formulation for finding a capacitated popular mixed matching. □

5.4. Two-sided preferences

Finally we consider the case where both applicants and jobs have preferences over the other side. The case of two-sided preference lists has been extensively studied in the literature [10,20]. It is known [9] that when preference lists are strict, a stable matching is popular but when preference lists contain ties the instance may admit no popular matching.

Let $G(\mathcal{A} \cup \mathcal{J}, E)$ be an instance of the popular matchings problem with two-sided preference lists. For two matchings M and T in G, we define $\phi(M, T)$ as below.

$$\phi(M, T) = |\{u \in \mathcal{A} \cup \mathcal{J} : u \text{ prefers } M \text{ over } T\}| .$$

Recall that, for the case of one-sided preference lists we introduced a unique last-resort job ℓ_a at the end of each applicant's preference list. Here in addition to that, we introduce a unique last resort applicant ℓ_j at the end of each job's preference list. We also connect every pair of last-resort job and last-resort applicant with an edge. This allows us to restrict our attention to complete matchings in G. The definition of popular matchings is exactly the same as in the case of one-sided preferences.

The set $\mathcal{X}$ of fractional matchings in G is defined as follows:

$$\mathcal{X} = \left\{ \vec{x} \in \mathbb{R}^m_+ \;\middle|\; \begin{array}{l} \sum_{e \in E(a)} \vec{x}(e) = 1 \text{ for } a \in \mathcal{A} \text{ and} \\ \sum_{e \in E(j)} \vec{x}(e) = 1 \text{ for } j \in \mathcal{J}. \end{array} \right\}$$

Note that $\mathcal{X}$ is integral. We now define the function Δ for fractional matchings $\vec{x_1}, \vec{x_2} \in \mathcal{X}$.

$$\Delta(\vec{x_1}, \vec{x_2}) = \sum_{u \in \mathcal{A} \cup \mathcal{J}} \sum_{\substack{(u,i) \in E(u) \\ (u,j) \in E(u)}} \vec{x_1}(u, i)\, \vec{x_2}(u, j)\, \mathrm{vote}_u(i, j) \tag{21}$$

where $\mathrm{vote}_u(i, j)$ is the natural extension of (10) to any $u \in \mathcal{A} \cup \mathcal{J}$.

Lemma 4. *Let P and Q be two mixed matchings for an instance G with two-sided preferences and let $\vec{x_1}$ and $\vec{x_2}$ be their corresponding fractional matchings. Then $\Delta(P, Q) = \Delta(\vec{x_1}, \vec{x_2})$.*

We omit the proof of Lemma 4 as it is identical to the proof of Lemma 1.

Theorem 7. *A two-sided popular mixed matching always exists and can be computed efficiently.*

Proof. We define the polyhedral game for the polytope $\mathcal{X}$ using the payoff matrix Δ. Because the payoff matrix is skewed symmetric the value of the game is zero and thus there always exists a two-sided popular mixed matching.

To compute such a mixed matching we note that the payoff matrix of the game defined by (21) is bilinear. Thus we can apply Theorem 4 to obtain a compact linear programming formulation for finding a popular mixed matching. □

6. Concluding remarks

In this paper we introduced the concept of popular mixed matchings and showed that they enjoy the best qualities of previously proposed solution concepts based on the popularity criterion; namely, they always exist and can be computed efficiently. With our approach we are able to tackle more general settings where applicants have priorities or weights [18], where multiple copies of the jobs are available [16], where we have preferences on both sides [9].

At the heart of our proofs is an interesting connection between popular matchings and the Nash equilibria of a certain zero-sum game with exponentially-many strategies. In general, finding these equilibria is equivalent to linear programming[3]; in our case, however, by exploiting problem-specific properties we could find an equivalent linear program whose size is polynomial on the size of the input graph. The same technique works for more general zero-sum games. Namely, those games whose strategies correspond to the extreme points of a polytope $\mathcal{P}$ and whose payoff matrix $S(\vec{x}, \vec{y})$ can be expressed as a bilinear function of $\vec{x}$ and $\vec{y}$ in $\mathcal{P}$.

We leave as an open problem to design purely combinatorial algorithms for computing popular mixed matchings.

Acknowledgements

We thank Naveen Garg for helpful discussions, Khaled Elbassioni for suggesting the generalization to bilinear payoffs, and Debasis Mishra for asking us about popular fractional assignments.

3 Linear programming can be reduced to the problem of solving a zero-sum game of roughly the same size [7, Chapter 13].

Appendix. Calculations from Section 2

Recall that our instance has applicants $\{a_1, a_2, b_1, b_2, x\}$ and jobs $\{i_1, i_2, j_1, j_2\}$ plus last resort jobs. The preferences lists are strict and are shown below

a_1	j_1	j_2	ℓ_{a_1}
a_2	j_1	j_2	ℓ_{a_2}
b_1	i_1	i_2	ℓ_{b_1}
b_2	i_1	i_2	ℓ_{b_2}
x	j_1	i_2	ℓ_x

The matching where all top jobs are iteratively shared equally among the applicants is equivalent to the uniform distribution over the following matchings

$$\begin{aligned}
T_1 &= \left\{(a_1, \ell_{a_1}),\ (a_2, j_2),\ (x, j_1),\ (b_1, i_1),\ (b_2, i_2)\right\},\\
T_2 &= \left\{(a_1, j_2),\ (a_2, \ell_{a_2}),\ (x, j_1),\ (b_1, i_2),\ (b_2, i_1)\right\},\\
T_3 &= \left\{(a_1, j_1),\ (a_2, j_2),\ (x, i_2),\ (b_1, i_1),\ (b_2, \ell_{b_2})\right\},\\
T_4 &= \left\{(a_1, j_2),\ (a_2, j_1),\ (x, i_2),\ (b_1, \ell_{b_1}),\ (b_2, i_1)\right\},\\
T_5 &= \left\{(a_1, j_1),\ (a_2, j_2),\ (x, \ell_x),\ (b_1, i_1),\ (b_2, i_2)\right\},\\
T_6 &= \left\{(a_1, j_2),\ (a_2, j_1),\ (x, \ell_x),\ (b_1, i_2),\ (b_2, i_1)\right\}.
\end{aligned}$$

In other words, we are working the mixed matching $Q = \left\{(T_k, \frac{1}{6}) : k = 1, \ldots, 6\right\}$. This is not popular as the matching

$$M_0 = \left\{(a_1, j_1),\ (a_2, j_2),\ (x, i_2),\ (b_1, i_1),\ (b_2, \ell_{b_2})\right\}$$

is more popular than Q. To prove this claim we need to compare $\phi(Q, M_0)$ to $\phi(M_0, Q)$.

$$\phi(Q, M_0) = \frac{\sum_{k=1}^{6} \phi(T_k, M_0)}{6} = \frac{2+2+0+2+1+2}{6} = \frac{9}{6}$$

$$\phi(M_0, Q) = \frac{\sum_{k=1}^{6} \phi(M_0, T_k)}{6} = \frac{1+3+0+2+1+3}{6} = \frac{10}{6}.$$

Since $\phi(M_0, Q) > \phi(Q, M_0)$ it follows that M_0 is more popular than Q.

References

[1] T.S.A. Abdulkadiroğlu, Ordinal efficiency and dominated sets of assignments, Journal of Economic Theory 112 (2003) 157–172.
[2] A. Abdulkadiroğlu, T. Sönmez, Random serial dictatorship and the core from random endowments in house allocation problems, Econometrica 66 (3) (1998) 689–701.
[3] D.J. Abraham, R.W. Irving, T. Kavitha, K. Mehlhorn, Popular matchings, SIAM Journal on Computing 37 (4) (2007) 1030–1045.
[4] A. Bogomolnaia, H. Moulin, A new solution to the random assignment problem, Journal of Economic Theory 100 (2) (2001) 295–328.
[5] A. Bogomolnaia, H. Moulin, A simple random assignment problem with a unique solution, Economic Theory 75 (3) (2004) 257–279.
[6] C. Carathéodory, Über den variabilitätsbereich der fourierschen konstanten von positiven harmonischen funktionen, Rendiconti del Circolo Matematico di Palermo 32 (1911) 193–217.
[7] G. Dantzig, Linear Programming and Extensions, Princeton University Press, 1998.
[8] J. Edmonds, R.M. Karp, Theoretical improvements in algorithmic efficiency for network flow problems, Journal of the ACM 19 (2) (1972) 248–264.
[9] P. Gardenfors, Match making: assignments based on bilateral preferences, Behavioural Sciences 20 (1975) 166–173.
[10] D. Gusfield, R.W. Irving, The Stable Marriage Problem: Structure and Algorithms, The MIT Press, 1989.
[11] C.-C. Haung, T. Kavitha, D. Michail, M. Nasre, Bounded unpopularity matchings. in: Proceedings of the 11th Scandinavian Workshop on Algorithm Theory, 2008, pp. 127–137.
[12] A.-K. Katta, J. Sethuraman, A solution to the random assignment problem on the full preference domain, Journal of Economic Theory 131 (1) (2005) 231–250.
[13] M. Mahdian, Random popular matchings, in: Proceedings of the 8th ACM Conference on Electronic Commerce, 2006, pp. 238–242.
[14] M. Manea, A constructive proof of the ordinal efficiency welfare theorem, Journal of Economic Theory 141 (2007) 276–281.
[15] M. Manea, Asymptotic ordinal inefficiency of random serial dictatorship, Theoretical Economics 4 (2) (2009) 165–197.
[16] D. Manlove, C. Sng, Popular matchings in the capacitated house allocation problem, in: Proceedings of the 14th Annual European Symposium on Algorithms, 2006, pp. 492–503.
[17] R.M. McCutchen, The least-unpopularity-factor and least-unpopularity-margin criteria for matching problems with one-sided preferences, in: Proceedings of the 15th Latin American Symposium on Theoretical Informatics, 2008, pp. 593–604.
[18] J. Mestre, Weighted popular matchings, in: Proceedings of the 33th International Colloquium on Automata, Languages and Programming, 2006, pp. 715–726.
[19] J. V. Neumann, Zur theorie der gesellschaftsspiele, Mathematische Annalen 100 (1928) 295–320.
[20] A.E. Roth, M.A.O. Sotomayor, Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis, Cambridge University Press, 1992.
[21] L. Zhou, On a conjecture by Gale about one-sided matching problems, Journal of Economic Theory 52 (1) (1990) 123–135.