AUPCR Maximizing Matchings : Towards a Pragmatic Notion of Optimality for One-Sided Preference Matchings

Girish Raguvir J[*], **Rahul Ramesh**[*], **Sachin Sridhar**[*], **Vignesh Manoharan**[*]
Department of Computer Science and Engineering
Indian Institute of Technology Madras, India

Abstract

We consider the problem of computing a matching in a bipartite graph in the presence of one-sided preferences. There are several well studied notions of optimality which include Pareto-optimality, rank-maximality, fairness and popularity. In this paper, we conduct an in-depth experimental study comparing different notions of optimality based on a variety of metrics like cardinality, number of rank 1 edges, popularity, to name a few. Observing certain shortcomings in the standard notions of optimality, we propose an algorithm which maximizes an alternative metric called the *Area under Profile Curve ratio* (AUPCR). To the best of our knowledge, the AUPCR metric was used earlier only as an evaluation metric, and there is no known algorithm to compute an AUPCR maximizing matching. Finally, we illustrate the superiority of the AUPCR-maximizing matching by comparing its performance against other optimal matchings on synthetic instances modeling real-world data.

Introduction

The problem of assigning elements of one set to elements of another set, is motivated by important real-world scenarios like assigning students to universities, applicants to jobs and so on. In many of these applications, members of one or both the sets rank each other in an order of preference. The goal is to compute an assignment that is "optimal" with respect to the preferences.

In this paper we focus on the *one-sided* preference list model, where members of one set rank a subset of elements in the other set in a linear order (preferences are assumed to be strict). Several notions of optimality like Pareto-optimality, rank-maximality, fairness and popularity have been considered in literature (we give formal definitions of each of these notions later). For each of the above mentioned notions of optimality, there are efficient algorithms studied in the literature to compute the specified optimal matching. Abraham et al. (2004) describe an algorithm that computes a maximum cardinality Pareto-optimal matching. Abraham et al. (2007) present an algorithm to compute a popular matching while Irving et al. (2004); Huang et al. (2016) propose algorithms that optimize the head/tail of the matching profile (rank-maximal and fair respectively). Maximizing one metric could however result in poor performance on other yardsticks of measure. When comparing two matchings, it is difficult to measure the quality of the two matchings using a single scalar value. They can be compared using a variety of metrics like cardinality, number of matched rank 1 edges or cardinality, none of which can serve as a sole indicator of optimality.

Profile based matchings, like rank-maximal or fair matchings which optimize for the head or the tail of the profile can turn out to be biased under certain circumstances. An alternative is to consider the *Area under Profile Curve Ratio* metric introduced by Diebold and Bichler (2017). This metric aims to maximize a measure, that is a weighted sum of matched edges, with the weight proportional to its position in the preference list.

In this work, we first present a comprehensive experimental study of the well-studied notions of optimality for matchings and compare them using different measures. We then describe the AUPCR metric, and propose algorithms to compute an AUPCR maximizing matching, and a maximum cardinality AUPCR maximizing matching.

Finally, we empirically evaluate different matching algorithms on synthetic graphs generated from generator models specified by Michail (2011) using various metrics. The generated graphs fall into two categories, one having uniformly random preference lists and the other having highly correlated preference lists. Our analysis is inspired by the analysis by Michail (2011), and we additionally consider a ranking system in which the matchings are ranked based on multiple metrics. These rankings are consequently aggregated to obtain a single rank, which we use as a coarse indicator of matching quality.

The AUPCR maximizing matching is experimentally shown to have good performance across evaluated metrics on the considered data-sets, and we believe this matching is well suited for practical applications.

Preliminaries

Consider a set $\mathcal{A}$ of applicants and a set $\mathcal{P}$ of posts. Every applicant a has a preference list over a subset of the posts in $\mathcal{P}$. This list is a linear order (strict list) and is called the preference list of a over $\mathcal{P}$. The problem is readily represented as a bipartite graph with vertices $\mathcal{V} = \mathcal{A} \cup \mathcal{P}$ and an edge (a, p) is present if p is exists in the preference list of a. Preferences of applicants are encoded by assigning ranks

[*] All authors contributed equally

to edges. Each edge (a, p) has a rank i if a considers p as its i-th most preferred post. A matching $M \subseteq E$ is a collection of edges such that no two edges share an endpoint. Let $|\mathcal{A} \cup \mathcal{P}| = n$ and $|E| = m$. We now define formally the different notions of optimality.

Maximum Cardinality Pareto-Optimal Matching

A matching M is said to be Pareto-optimal if there is no other matching M' such that some applicant is better off in M' while no applicant is worse off in M' than in M(an applicant is worse of in M if it is matched to a less preferred vertex compared to M') . Maximum cardinality Pareto-optimal matchings(POM) can be computed in $\mathcal{O}(m\sqrt{n})$ time using the algorithm given by Abraham et al. (2004).

Rank-Maximal Matching

The notion of *rank-maximal* matchings was first introduced by Irving under the name of *greedy matchings* (Irving, 2003). A rank-maximal matching is a matching in which the number of rank one edges is maximized, subject to which the number of rank two edges is maximized and so on. Another way of defining rank-maximal matchings is through their $signatures$. Given that r is the largest rank across all preference lists, we define the signature of a matching M as an $r+1$-tuple $(x_1, x_2, \cdots, x_r, x_{r+1})$ where, for $1 \leq i \leq r$, x_i represents the number of applicants matched to their i-th preference (x_{r+1} denotes the number of unmatched applicants).

Let $(x_1, x_2, \cdots, x_r, x_{r+1})$ and $(y_1, y_2, \cdots, y_r, y_{r+1})$ denote the signatures of M and M' respectively. We say that $M \succ M'$ w.r.t. rank-maximality if there exists an index k with $1 \leq k \leq r$ such that for $1 \leq i < k$, $x_i = y_i$, and $x_k > y_k$. A matching M is rank-maximal if M has the best signature w.r.t. rank-maximality.

Through the rest of the paper, we denote this matching as RMM. For the purposes of our experimental evaluation, we implement a simple combinatorial algorithm (Irving et al., 2004) to compute a rank-maximal matching. The running time of the algorithm is $O(\min(C\sqrt{n}, n + C) \cdot m)$ where C is the maximum rank of an edge in the matching.

Popular Matching

To define popularity, we translate preferences of applicants over posts to preferences of applicants over matchings. An applicant a prefers matching M to M' if either a is matched in M and unmatched in M', or a is matched in both M and M' but has better rank in M than in M'. A matching M is more popular than M' if the number of applicants who prefer M to M' exceeds the number of applicants who prefer M' to M. A matching M is popular if there is no matching that is more popular than M.

The more popular than relation is not transitive, and it is possible that a popular matching does not exist. Abraham et al. (2007) describe a linear-time algorithm for strict preferences that checks for the existence of a popular matching and computes the maximum cardinality popular matching if it exists. When a popular matching does not exist, one can attempt to minimize the unpopularity factor given by McCutchen (2008). The bounded unpopularity matching algorithm (Huang et al., 2011) finds a popular matching if it exists, and otherwise finds an approximation to the matching with the least unpopularity factor. This approximation ratio can however be as bad as $O(n)$ in the worst case. Through the rest of the paper, we denote the bounded unpopularity matching as POPM.

Fair Matching

Fair matchings can be considered as complementary to rank-maximal matchings. A fair matching is always a maximum cardinality matching; subject to this it matches the least number of applicants their last preferred post, subject to this, least number of applicants to their second last preferred post and so on. Fair matchings can be conveniently defined using signatures.

Let $(x_1, x_2, \cdots, x_r, x_{r+1})$ and $(y_1, y_2, \cdots, y_r, y_{r+1})$ denote the signatures of two matchings M and M' respectively. We say that $M \succ M'$ w.r.t. fairness if there exists a index k with $1 \leq k \leq r+1$ such that for $k < i \leq r+1$, $x_i = y_i$, and $x_k < y_k$. A matching is fair if it is of maximum cardinality, and subject to that it has the best signature according to the above defined criteria. Recently Huang et al. (2016) gave a combinatorial algorithm to compute fair matchings. Through the rest of the paper, we denote this matching as FM.

AUPCR Maximizing Matching

Fair and Rank-maximal matchings are profile based matchings that are geared towards minimizing the tail or maximizing the head of the profile. However, optimizing for the peripheral portions of a profile may not be necessarily representative of a *good* matching in many practical settings. This encouraged us to look into a metric called *Area Under Profile Curve Ratio* (AUPCR) which in a sense seemed to capture the entire signature of a matching.

Formulation of AUPCR

The *Area Under Profile Curve Ratio* (AUPCR), introduced under the context of matchings by Diebold and Bichler (2017) is a measure of second order stochastic dominance of the profile. It is a useful metric that can be used to compare multiple signatures and is very similar in nature to the highly popular Area Under Curve of Receiver Operating Characteristic (Hanley and McNeil, 1982).

For a matching M of a bipartite graph $G(\mathcal{A} \cup \mathcal{P}, E)$ with $n_i(M)$ representing the number of applicants matched to their i'th preference, AUPCR(M) is defined as the ratio of *Area Under Profile Curve* (AUPC) and *Total Area* (TA) where :

$$\text{TA}(M) = |\mathcal{A}||\mathcal{P}| \tag{1}$$

$$\begin{aligned}\text{AUPC}(M) &= \sum_{r=1}^{|\mathcal{P}|} |(a_i, p_j) \in M : rank(a_i, p_j) \leq r| \\ &= \sum_{i=1}^{|\mathcal{P}|} (|\mathcal{P}| - i + 1) n_i(M) \end{aligned} \tag{2}$$

Giving us,

$$\text{AUPCR}(M) = \frac{\sum_{i=1}^{|\mathcal{P}|}(|\mathcal{P}| - i + 1)n_i(M)}{|\mathcal{A}||\mathcal{P}|} \quad (3)$$

One can visualize this quantity by considering Figure 1. For an instance with $|\mathcal{A}| = 8$, $|\mathcal{P}| = 6$ and signature of matching M given by $(4, 0, 2, 1, 1, 0)$, the area under the shaded region corresponds to AUPC(M) ($= 4{+}4{+}6{+}7{+}8{+}8 = 37$) while the area of bounding rectangle corresponds to TA(M)($= 8 \times 6 = 48$). With these computed, AUPCR(M) is essentially the ratio of the two and is given by $\frac{37}{48} \approx 0.771$.

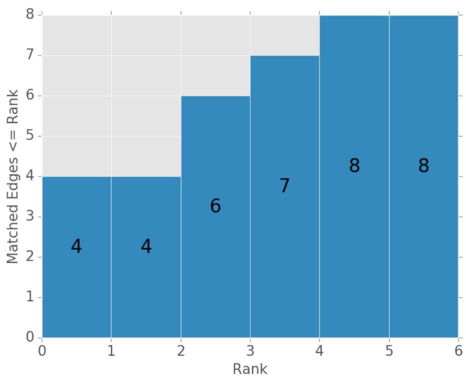


Figure 1: AUPCR - Visualization

A matching that maximizes this measure can be seen as a ”softer” version of the rank-maximal matching; it does not give up matching low ranked edges entirely in order to match a large number of high ranked edges. Based on this we consider two problems :

- AUPCR Maximizing Matching - the problem of finding a matching which maximizes the AUPCR metric. We denote such a matching as AMM.
- Max Cardinality AMM - the problem of finding a matching with the maximum cardinality among all matchings with maximum AUPCR. We denote such a matching as MC-AMM.

In this paper, we formulate algorithms to address the above defined problems and show that the AUPCR maximizing matching performs favourably on a variety of other standard metrics typically used to compare matchings in practical settings.

Algorithm - AUPCR Maximizing Matching

The problem of finding an AUPCR maximizing matching can be reduced to the problem of finding a maximum weighted perfect matching. Given a bipartite graph $G(\mathcal{A} \cup \mathcal{P}, E)$ and a weight w_e for each edge $e \in E$, we define the weight of a matching as $w(M) = \sum_{e \in E} w_e$. Then the maximum weighted perfect matching problem is to find a matching M which matches all vertices in $\mathcal{A}$ (M is a perfect matching) and maximizes $w(M)$.

Given a bipartite graph $G(\mathcal{A} \cup \mathcal{P}, E)$ with edges representing preferences of $\mathcal{A}$ (as described in the Preliminaries section), we construct $G'(\mathcal{A}' \cup \mathcal{P}', E')$ as follows:

1. $\mathcal{A}' = \mathcal{A}_1 \cup \mathcal{P}_2$ and $\mathcal{P}' = \mathcal{P}_1 \cup \mathcal{A}_2$ where $\mathcal{A}_1, \mathcal{A}_2$ are copies of $\mathcal{A}$ and $\mathcal{P}_1, \mathcal{P}_2$ are copies of $\mathcal{P}$.
2. For each edge $e \in E$ of rank i, add edge between corresponding vertices of $\mathcal{A}_1$ and $\mathcal{P}_1$ with weight $|\mathcal{P}| - i + 1$. Similarly, add an edge between $\mathcal{A}_2$ and $\mathcal{P}_2$ with the same weight.
3. Add edges with weight 0 from vertices in $\mathcal{A}_1$ to their copies in $\mathcal{A}_2$. Add similar edges between $\mathcal{P}_1$ and $\mathcal{P}_2$. We refer to these edges as identity edges.

Proof of Correctness

Claim. If M is a max weighted perfect matching in G', then M restricted to $\mathcal{A}_1 \cup \mathcal{P}_1$ is a AMM in G.

Proof. Let M_1 and M_2 be the matchings obtained by restricting M to $\mathcal{A}_1 \cup \mathcal{P}_1$ and $\mathcal{A}_2 \cup \mathcal{P}_2$ respectively, excluding the identity edges in both cases. Since M is a perfect matching, if a vertex in $\mathcal{A}_1 \cup \mathcal{P}_1$ is not matched in M_1, it must be matched to its copy in $\mathcal{A}_2 \cup \mathcal{P}_2$ via the identity edge. This means that its copy is also unmatched in M_2. This means that M_1 and M_2 match the same set of vertices. Since the identity edges have 0 weight,

$$w(M) = w(M_1) + w(M_2)$$

Since M_1 and M_2 match the same set of vertices, one can copy the edges matched in M_1 to $\mathcal{A}_2 \cup \mathcal{P}_2$. This means that $w(M_1) = w(M_2)$ and $w(M) = 2w(M_1)$. Maximizing $w(M)$ is equivalent to maximizing $w(M_1)$.

We also have

$$w(M_1) = \sum_{e \in M_1} w_e = \sum_{e \in M_1} |\mathcal{P}| - r_e + 1$$
$$= \sum_{i=1}^{|\mathcal{P}|} n_i(|\mathcal{P}| - i + 1) = |\mathcal{A}||\mathcal{P}|\text{AUPCR}(M_1)$$

where r_e is the rank of edge e and n_i is the number of edges of M_1 with rank i.This means that maximizing $w(M_1)$ maximizes AUPCR(M_1).

Hence, if M is a maximum weight perfect matching in G', M_1 is a max AUPCR matching in G. □

Algorithm - Max Cardinality AMM

The problem of finding a Max Cardinality AMM can also be reduced to an instance of max weighted perfect matching. The reduction is the same as the max AUPCR case, expect that we add a negative weight of $-\frac{1}{|\mathcal{A}|+|\mathcal{P}|}$ to the identity edges going from $\mathcal{A}_1$ to $\mathcal{A}_2$ in the graph G'.

Proof of Correctness

Claim. If M is a max weighted perfect matching in G', then M restricted to $\mathcal{A}_1 \cup \mathcal{P}_1$ is a Max Cardinality AMM in G.

Proof. As before, we can prove that $w(M_1) = w(M_2)$. However, $w(M) = w(M_1) + w(M_2) + w(I)$ where I is the set of identity edges in M. If M_1 leaves k_A vertices in $\mathcal{A}_1$ and $k_\mathcal{P}$ vertices in $\mathcal{P}_1$ unmatched, then M_2 also leaves the same vertices unmatched. So, we have $2(k_A + k_\mathcal{P})$ identity edges in I and hence

$$w(I) = -2\frac{k_A + k_\mathcal{P}}{|\mathcal{A}| + |\mathcal{P}|}$$

Since $k_A + k_{\mathcal{P}} < |\mathcal{A}| + |\mathcal{P}|$, we have $-2 < w(I) \le 0$ and

$$2(w(M_1) - 1) < w(M) \le 2w(M_1)$$

Let M_1' be a max AUPCR matching on $\mathcal{A}_1 \cup \mathcal{P}_1$, and M' be its extension to G' by copying edges and adding identity edges to match the remaining vertices. Since M is a maximum weighted perfect matching, $w(M') \le w(M)$ and so

$$\begin{aligned} &2(w(M_1') - 1) < 2w(M_1) \\ \Rightarrow &w(M_1') - w(M_1) < 1 \\ \Rightarrow &|\mathcal{A}||\mathcal{P}|\text{AUPCR}(M_1') - |\mathcal{A}||\mathcal{P}|\text{AUPCR}(M_1) < 1 \\ \Rightarrow &\text{AUPCR}(M_1') - \text{AUPCR}(M_1) < \frac{1}{|\mathcal{A}||\mathcal{P}|} \end{aligned}$$

From the definition of AUPCR, we can see that if two matchings have different AUPCR, then the difference is $\ge \frac{1}{|\mathcal{A}||\mathcal{P}|}$. So, M_1' and M_1 have the same AUPCR, which means that M_1 is an AUPCR maximizing matching in G.

The cardinality of M_1 is $|M_1| = |\mathcal{A}| - k_A = |\mathcal{P}| - k_{\mathcal{P}}$. Writing $w(M)$ in terms of $|M_1|$,

$$w(M) = 2w(M_1) - \frac{|\mathcal{A}| + |\mathcal{P}| - 2|M_1|}{|\mathcal{A}| + |\mathcal{P}|}$$

All AUPCR maximizing matchings will have the same $w(M_1)$, which means that maximizing $w(M)$ maximizes $|M_1|$. So, M_1 is a maximum cardinality AUPCR maximizing matching in G. □

The time complexity of the algorithm to find maximum weighted matching presented is $O(m\sqrt{n}\log n)$ (Duan and Su, 2012). Since both our algorithms construct a graph with $2n$ vertices and $2m + n$ edges and find a max weighted matching, the time complexity for finding both AMM and MC-AMM would be $O((m + n)\sqrt{n}\log n)$. A comparison of the run-time complexities is presented in Table 1.

Algorithm	Running Time
POM (Abraham et al., 2004)	$\mathcal{O}(m\sqrt{n})$
RMM (Irving et al., 2004)	$\mathcal{O}(\min(C\sqrt{n}, n + C)m)$
FM (Huang et al., 2016)	$\mathcal{O}(Cm\sqrt{n}\log n)$
POPM (Huang et al., 2011)	$\mathcal{O}(m\sqrt{n})$
AMM, MC-AMM (Our work)	$\mathcal{O}((m + n)\sqrt{n}\log n)$

Table 1: C is the maximum rank of any edge in the matching, $n = |A| + |P|$ and $m = |E|$

Understanding AMM

The formulation of AMMs lead us to explore some interesting questions.

Is an AMM Pareto-optimal?

Yes, AMM is a Pareto-optimal matching.

Theorem. *AUPCR maximizing matching is Pareto-optimal.*

Proof. Assume on the contrary that an AUPCR maximizing matching M is not Pareto-optimal. This means there exists a matching M' where every applicant in M' is at least as well off as in M and at least one applicant in M' is better off than M. Consider a vertex $v \in \mathcal{A}$. Let $r_M(v)$ be the rank of the post that v is matched to ($r_M(v) = |\mathcal{P}| + 1$ if v is unmatched), and $r_{M'}(v)$ be defined analogously.

$$\begin{aligned} &AUPCR(M') - AUPCR(M) \\ &= \sum_{v \in \mathcal{A}} ((\mathcal{P} - r_{M'}(v) + 1) - (\mathcal{P} - r_M(v) + 1)) \\ &= \sum_{v \in \mathcal{A}} (r_{M'}(v) - r_M(v)) \\ &> 0 \end{aligned}$$

The last inequality follows from the fact that every term of the summation is non negative and at least one term is positive by our assumption that M is not Pareto-optimal.
Since AUPCR(M') - AUPCR(M) > 0, M is not an AUCPR maximizing matching, a contradiction, and so M must be Pareto-optimal. □

Is an AMM always a maximum cardinality matching?

An AMM need not always be a maximum cardinality matching. Consider the instance with $\mathcal{A} = \{a_1, a_2, a_3, a_4\}$, $\mathcal{P} = \{b_1, b_2, b_3, b_4\}$ and the preferences given by

$$\begin{aligned} a_1 &: (b_1, 1) \\ a_2 &: (b_1, 1), (b_2, 2) \\ a_3 &: (b_2, 1), (b_1, 2), (b_3, 3) \\ a_4 &: (b_3, 1), (b_1, 2), (b_4, 3) \end{aligned}$$

As shown in Figure 2 and Figure 3, for this instance, AMM has a cardinality of 3 while a maximum cardinality matching has a cardinality of 4.

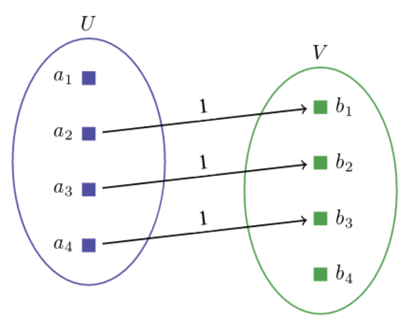


Figure 2: AUPCR Maximizing Matching

Do all AMMs have the same cardinality?

All AMMs need not have the same cardinality. Consider the instance with $\mathcal{A} = \{a_1, ...a_6\}$ and $\mathcal{P} = \{b_1, ...b_6\}$ and the preferences given by

$$\begin{aligned} a_1 &: (b_6, 1), (b_3, 2), (b_1, 3) \\ a_2 &: (b_2, 1), (b_3, 2), (b_1, 3) \\ a_3 &: (b_4, 1), (b_5, 2), (b_2, 3) \\ a_4 &: (b_1, 1), (b_4, 2), (b_6, 3) \\ a_5 &: (b_5, 1), (b_2, 2), (b_1, 3) \\ a_6 &: (b_4, 1), (b_2, 2), (b_5, 3) \end{aligned}$$

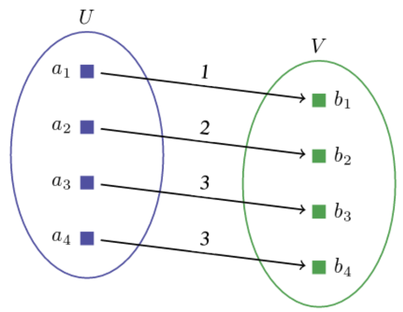


Figure 3: Maximum Cardinality Matching

As shown in Figure 4 and Figure 5, both are AUPCR maximizing matchings, with an AUPCR of 0.833, but they have different cardinalities. This example also shows that multiple AMMs can exist for a given instance.

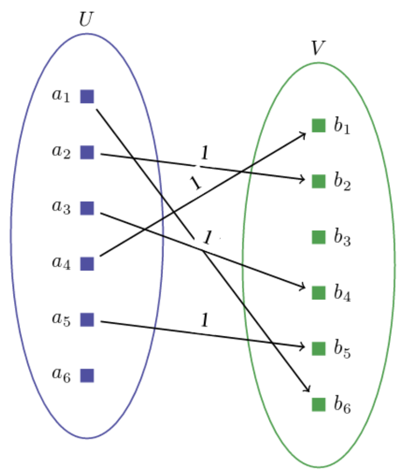


Figure 4: An AMM with $|M| = 5$

Experiments

To compare AMMs with other matchings, we consider a set of the evaluation metrics which we believe characterizes preference matchings and are of importance in practice. Some of these choices are inspired from the work of Michail (2011).

Evaluation Metrics

The matchings obtained from each algorithm are evaluated with respect to the following metrics.

- **Cardinality**: Number of edges present in the matching.
- **Unpopularity measure**: The unpopularity measure $u(M, M')$ measures how far away a matching M is from a popular(least unpopular) matching M'. Let $p(M_1, M_2)$ be the number of applicants that prefer M_1 over M_2. Then $u(M, M')$ for matching M is defined as the ratio of $p(M', M) - p(M, M')$ to the total number of applicants.
- **Rank 1**: The number of matched *rank 1* edges

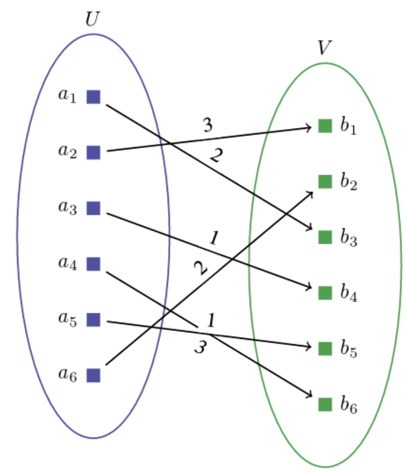


Figure 5: An AMM with $|M| = 6$

- **AUPCR**: The AUPCR metric is second order stochastic dominance of the profile as defined in Equation 3.
- **Ranks less than half the preference list size (RHPL)**: This counts the number of applicants who have been matched to a post with a rank better than or equal to half the length of their preference list.
- **Average rank**: For a matching M, this is the average rank of all matched edges. Although this is similar to the AUPCR metric, the average rank is computed only over the matched edges while AUPCR accounts for unmatched edges.
- **Worst rank**: For a matching M, this is the highest (worst) rank among all matched edges in M.

Instances

For our experiments, we consider two structured instance generators, *Highly Correlated* and *Uniform Random*. These generators are identical in nature to those given by Michail (2011), but we consider only instances with strict preference lists. Though all the algorithms described above (except Maximum Cardinality Pareto-Optimal) can also handle instances with ties, we went with this choice to have a set of instances upon which all the algorithms could be compared and analyzed. This choice is not too restrictive as in practical scenarios preference lists are often strict and devoid of ties.

Uniform Random (UNI) UNI instances are parameterized by a density d with $0 \leq d \leq 1$. Every applicant has a preference list size of $l = \lfloor |\mathcal{P}| \cdot d \rfloor$. These preference lists are chosen uniformly at random from the set of permutations of l posts. Let an applicant a's adjacency list be $(p_1, p_2, ..., p_l)$. Then p_1 is ranked 1 by a, p_2 is ranked 2, and so on.

Highly Correlated (HC) These instances are generated based on a global preference ordering(say p_g) for the set of posts; one that all the applicants agree upon. Similar to UNI, a HC instance is also parameterized by a density d with $0 \leq d \leq 1$. For every vertex pair (u, v) with $u \in \mathcal{A}, v \in \mathcal{P}$, an edge (u, v) is added with a probability d. Once the graph has been constructed, the applicants rank the posts as per the global preference list: the best post, as per p_g, an applicant is connected to is assigned rank 1, and so on. Unlike UNI, preference list need not have identical lengths across all applicants.

Experimental Setup

The number of applicants are equal to the number of posts in any graph and is varied from 50 to 900 in steps of 50. Orthogonally, the density parameter d for HC and UNI is varied from 0.02 to 0.20 in steps 0.02. The reason for this choice of range is that real world datasets are not very dense in nature. Each instance is averaged over 50 random seeds. There exists one more level of averaging across different density (d) values to get one value for each metric for each problem size (number of applicants).

The variant of Max AUPCR that does not not enforce maximum cardinality is used. Surprisingly this still yields a max cardinality matching without exception. For POPM, in cases where popular matchings do not exist, the least unpopular matching is utilized. The code was executed using the Amazon web services (AWS) based EC2 service on a t2.micro instance (1 GB Ram, 1 CPU, Intel Xeon processor).

In general, POP, RMM, POPM, FM and AMM are not unique and the reported results are for the matchings obtained by the respective algorithms.

Experimental Results

	POM	RMM	POPM	FM	AMM
Card.	**1.00**	2.94	2.00	**1.00**	**1.00**
Unpop.	5.00	2.00	**1.00**	3.99	3.01
Rank 1	3.80	**1.00**	**1.00**	3.18	2.00
AUPCR	3.81	4.85	3.28	1.99	**1.00**
RHPL	4.28	2.89	2.58	1.26	**1.14**
Avg Rank	4.98	2.58	3.81	2.34	**1.22**
Worst Rank	4.66	3.27	3.39	**1.00**	1.94
Rank Mean	3.93	2.79	2.44	2.11	**1.62**

Table 2: Rank means of algorithms on metrics for UNI

Comparing matchings based on rank means
In this analysis, for a given metric and graph instance, we rank the algorithms in terms of performance with the best one getting a rank of 1 and worst one getting a rank of 5. We then average this rank across all instances and this value corresponds to an entry in Table 2. The *rank mean* is computed by taking the average of the entries along the column. This value is intended to serve as a measure of overall performance.

As seen from the table, each chosen metric has a subset of the algorithms performing best. It is however important to note that AMM performs competitively in almost all metrics. This observation is also qualitatively supported from the fact that the *rank mean* attained by AMM is lowest among all algorithms for both UNI and HC instances. This empirically shows that AMM is able to achieve a much desired balance, making it a very compelling choice for many practical preference matching problems.

	POM	RMM	POPM	FM	AMM
Card.	**1.00**	2.89	2.11	**1.00**	**1.00**
Unpop.	5.00	**3.07**	**1.00**	3.94	2.00
Rank 1	4.93	**1.00**	1.59	2.99	3.96
AUPCR	3.25	4.84	3.92	2.00	**1.00**
RHPL	4.09	1.96	3.02	4.75	**1.11**
Avg Rank	5.00	2.18	3.53	3.20	**1.09**
Worst Rank	3.66	4.73	3.42	**1.01**	1.99
Rank Mean	3.84	2.95	2.66	2.69	**1.73**

Table 3: Rank means of algorithms on metrics for HC

A graphical comparison for some of these metrics can be found in Figure 6 and Figure 7.

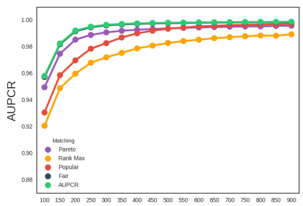
Observations

Some interesting observations we made are as follows :

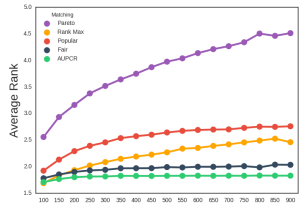
- **Cardinality** : As expected, POM and FM have the largest cardinality since they compute maximum cardinality matchings. However, it was observed that AMM without exception returned a maximum cardinality matching. While this may not universally true (as proved in consequent section) this is a useful property in practice.
- **RHPL** : The RHPL is one metric that no matching in particular optimizes for. It is peculiar to note that AUPCR maximizes this metric indicating that it is indeed a more general notion of optimality.
- **Rank 1** : It was observed that both popular and rank-maximal matchings have similar if not same number of rank 1 edges. While the head of the signature is maximized, it is observed that both these matchings display poor performances on metrics that account for the entirety or the tail of the signature.
- **Time** : Dictated by the computational time complexities of the respective algorithms, the times were vastly different for FM and AMM compared to the other three matchings. In graphs with 900 vertices (in each partition), the FM took 512.45 seconds, AMM executed in 204.78 seconds while POPM and POM were executed in less than 5 seconds.

Observing Table 2 and Table 3, one question which comes to mind is

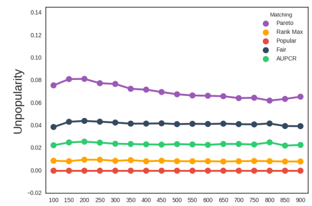
Is an AMM always more ”rank-maximal” than a FM?
No, an AMM matching may not always be more rank-maximal than the fair matching. Consider the instance with $\mathcal{A} = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7\}$, $\mathcal{P} =$



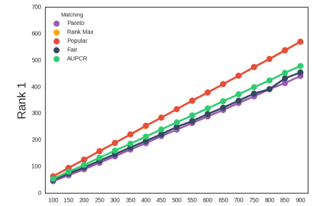
(a) AUPCR



(b) Avg Rank

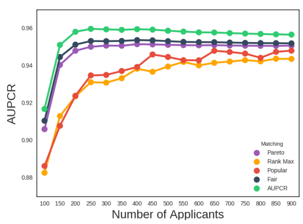


(c) Popularity

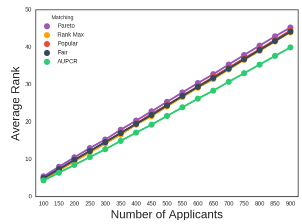


(d) Rank 1

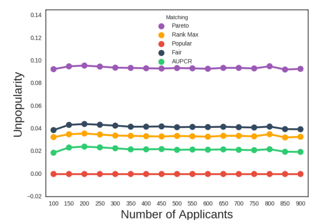
Figure 6: Uniform Random



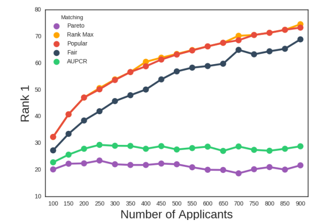
(a) AUPCR



(b) Avg Rank



(c) Popularity



(d) Rank 1

Figure 7: Highly Correlated

$\{b_1, b_2, b_3, b_4, b_5, b_6, b_7\}$ and the preferences given by

$$a_1 : (b_1, 1)$$
$$a_2 : (b_2, 1)$$
$$a_3 : (b_3, 1), (b_4, 2),$$
$$a_4 : (b_1, 1), (b_5, 2), (b_4, 3)$$
$$a_5 : (b_1, 1), (b_6, 2), (b_2, 3), (b_5, 4)$$
$$a_6 : (b_1, 1), (b_2, 2), (b_7, 3), (b_6, 4), (b_3, 5)$$
$$a_7 : (b_7, 1)$$

An AMM matching for the above graph is as show in Figure 8 and its signature is given by $(3, 3, 0, 0, 1)$. One can easily see that the FM shown in Figure 9 is more rank-maximal with a signature $(4, 0, 1, 2, 0)$.

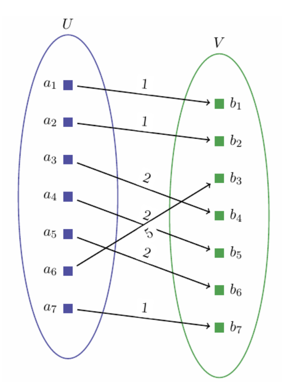


Figure 8: An AMM with matching with signature $(3, 3, 0, 0, 1)$

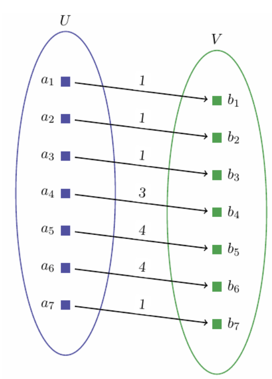


Figure 9: A Fair matching with signature $(4, 0, 1, 2, 0)$

Conclusion

In this work, we introduce the notion of an AUPCR maximizing matching. We describe two variants with one maximizing the AUPCR, and the other maximizing the cardinality subject to maximizing the AUPCR. We empirically evaluate our algorithm on standard synthetically generated datasets and highlight that AUPCR maximizing matchings achieve this much needed middle-ground with respect to the different notions of optimality. The overall performance of the AUPCR matching is superior in comparison to other matchings when all metrics are cumulatively used for comparison. Extending AUPCR matchings and finding algorithms with reduced time complexity is left as future work.

References

Abraham, D. J.; Cechlárová, K.; Manlove, D. F.; and Mehlhorn, K. 2004. Pareto optimality in house allocation problems. In *International Symposium on Algorithms and Computation*, 3–15. Springer.

Abraham, D. J.; Irving, R. W.; Kavitha, T.; and Mehlhorn, K. 2007. Popular matchings. *SIAM Journal on Computing* 37(4):1030–1045.

Diebold, F., and Bichler, M. 2017. Matching with indifferences: A comparison of algorithms in the context of course allocation. *European Journal of Operational Research* 260(1):268–282.

Duan, R., and Su, H.-H. 2012. A scaling algorithm for maximum weight matching in bipartite graphs. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, 1413–1424. SIAM.

Hanley, J. A., and McNeil, B. J. 1982. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology* 143(1):29–36.

Huang, C.-C.; Kavitha, T.; Michail, D.; and Nasre, M. 2011. Bounded unpopularity matchings. *Algorithmica* 61(3):738–757.

Huang, C.-C.; Kavitha, T.; Mehlhorn, K.; and Michail, D. 2016. Fair matchings and related problems. *Algorithmica* 74(3):1184–1203.

Irving, R. W.; Kavitha, T.; Mehlhorn, K.; Michail, D.; and Paluch, K. 2004. Rank-maximal matchings. In *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, 68–75. Society for Industrial and Applied Mathematics.

Irving, R. W. 2003. Greedy matchings. *Technical R eport Tr-2003-136, University of G lasgow*.

McCutchen, R. 2008. The least-unpopularity-factor and least-unpopularity-margin criteria for matching problems with one-sided preferences. *LATIN 2008: Theoretical Informatics* 593–604.

Michail, D. 2011. An experimental comparison of single-sided preference matching algorithms. *Journal of Experimental Algorithmics (JEA)* 16:1–4.