MatchXplain: Analyzing Preferences, Explaining Outcomes, and Simplifying Decisions

Hadi Hosseini[1] , **Yubo Jing**[2] and **Ronak Singh**[1]
[1]Pennsylvania State University
[2]Brown University
{hadi, ronak.singh}@psu.edu, yubo_jing@brown.edu

Abstract

Matching markets, where agents are assigned to one another based on preferences and constraints, are fundamental in various AI-driven applications such as school choice, content matching, and recommender systems. A key challenge in these markets is understanding preference data, as the interpretability of algorithmic solutions hinges on accurately capturing and explaining preferences. We introduce MatchXplain, a platform that integrates preference explanation with a robust matching engine. MatchXplain offers a layered approach for explaining preferences, computing diverse matching solutions, and providing interactive visualizations to enhance user understanding. By bridging algorithmic decision-making with explainability, MatchXplain improves transparency and trust in algorithmic matching markets.

1 Introduction

Matching markets are fundamental for facilitating pairings between two distinct groups based on ranked preferences. These markets play a central role in various real-world applications such as college admission and medical residency, and led to development of seminal algorithms to ensure their stability [Gale and Shapley, 1962]. Since their introduction, matching markets have became essential in a variety of fields beyond college admissions, including job placement [Roth and Peranson, 1999], school admissions [Abdulkadiroğlu *et al.*, 2005a; Abdulkadiroğlu *et al.*, 2005b], recommender systems [Eskandanian and Mobasher, 2020], electric vehicle charging [Gerding *et al.*, 2013], and ride sharing [Banerjee and Johari, 2019]. The effectiveness and long-term success of these markets hinges on several key properties including *stability*, *economic efficiency*, and their *fairness* guarantees. Each of these properties is critical in ensuring fairness and avoiding conflict in the market. Among the algorithms designed to address these challenges, perhaps the most-recognized mechanism is the *Deferred Acceptance* (DA) algorithm. Given a set of ordinal preferences, the DA algorithm is guaranteed to find a stable matching [Gale and Shapley, 1962], i.e., a solution wherein no pair of agents prefer each other to their matched partners.

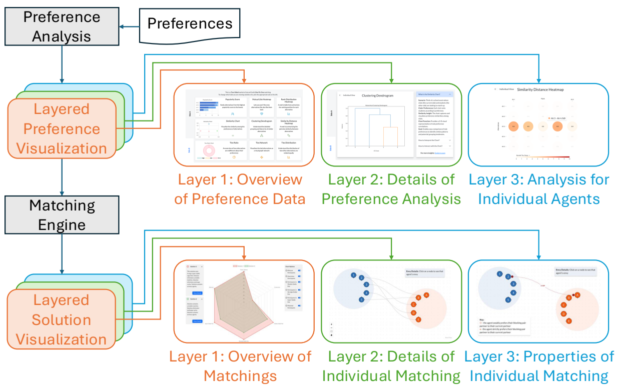

Figure 1: The platform's multi-layered user flow begins with the input of a preference profile. MatchXplain then analyzes the data to extract insights, which the matching engine uses to generate multiple matching options. Finally, users can select an option, explore it visually, and interact with the solution.

The theoretical guarantees of DA hold under strict and complete linear preferences. However, when preferences include ties or are incomplete, challenges arise regarding matching size, efficiency, stability type, and fairness. While various algorithms have been developed to address specific aspects of these challenges [Gusfield, 1987], no single solution satisfies all desirable properties across all problem instances. Moreover, inherent trade-offs—such as the incompatibility between Pareto optimality and stability [Abdulkadiroğlu *et al.*, 2020]—as well as computational complexity issues [Manlove *et al.*, 2002] make a universal, one-size-fits-all solution unattainable.

MatchXplain (www.matchxplain.org) aims to inform and guide the decisions of social planners and policymakers. To this end, it offers a menu of recommended solutions—derived from input preferences—that satisfy different sets of properties, allowing decision-makers to explore, adjust, and better understand potential outcomes. Effective explanation of an outcome requires a deep understanding of the nuances within the input data. MatchXplain bridges the gap between inputs and outputs by providing a comprehensive set of analytics and visualizations on the input preferences in a layered approach, enhancing interpretablity of decisions.

Related Platforms. With the recent wide-spread attention to algorithmic AI solutions, several platforms now offer provable and accessible algorithms for collective decision-making. For example, *MatchU.ai* [Ferris and Hosseini, 2020] emphasizes interactive visualization for pedagogical purposes; MATWA [Glitzner and Manlove, 2024] provides a web toolkit of matching algorithms. Broader platforms like *Spliddit.org* [Goldman and Procaccia, 2015] address fair resource allocation; *Panelot.org* [Flanigan *et al.*, 2021] focuses on fair civic lotteries—all related to matching problems under preferences. Our platform distinguishes itself by emphasizing preference mapping and visualization, solution explanation, and interactive tools for understanding desired properties.

2 Model and Properties

Matching Model. A two-sided matching market consists of two disjoint sets, often denoted by A and B, where each agent i has a preference list over the members of the other side. The preference list of agent i is a weak and potentially partial order, and is denoted by $\succeq_i$. We use $b_1 \succ_a b_2$ to denote that agent a prefers b_1 to b_2. We additionally use $b_1 \succeq_a b_2$ to denote that agent a weakly prefers b_1 to b_2 (a may prefer b_1 and b_2 equally). In the case of incomplete preferences, we use $\phi \succ_a b_1$ to denote that b_1 does not appear in a's preference list. A *preference profile* is simply the collection of all preferences lists of agents in both sides of the market, and is denoted by $\succ = (\succ_{a_1}, \ldots, \succ_{a_m}, \succ_{b_1}, \ldots, \succ_{b_n})$, where $m = |A|$ and $n = |B|$.

A *matching* is a mapping $\mu : A \cup B \to A \cup B$ such that $|\mu(a)| \leq 1$ and $\mu(a) \subseteq B$ for all $a \in A$, $|\mu(b)| \leq c_b$ (where c_b is the maximum number of agents that b can be matched with) and $\mu(b) \subseteq A$ for all $b \in B$, and $b \in \mu(a)$ if and only if $a \in \mu(b)$. For now, we focus on the special case of one-to-one matchings, where capacities c_b for all agents $b \in B$ are set to 1.

Properties. Given a matching μ, a *blocking pair* is a pair (a, b) who prefer each other over their assigned match, i.e., $b \succ_a \mu(a)$ and $a \succ_b \mu(b)$. A matching is said to be *stable* if it does not have any blocking pairs. Stability is often seen as a *fairness* notion under the name of 'justified envy-freeness'. It states that any envy that agent a_1 may feel for a_2's match, say b_1, is "justified" by agent b_1 preferring a_2 to a_1, that is, $a_2 \succ_{b_1} a_1$. A matching is *non-wasteful* if no unmatched $a_i \in A$ can be matched without changing the partner of at least one $a_j \in A$ (same condition applies for the set B). In the one-to-many setting, a stable solution is defined as one which satisfies both justified envy-freeness and non-wastefulness.

A matching μ is *Pareto-optimal* relative to set A if no $a_i \in A$ can strictly improve their assigned match without rematching at least one $a_j \in A$ with a strictly less-preferred partner (same condition applies for set B). The size of a matching μ is the number of individuals pairs (a, b) in the matching, where $\mu(a) = \{b\}$ and $\mu(b) = \{a\}$.

3 General Framework

The MatchXplain platform is comprised of three distinct but related components: MatchXplain's preference analysis, MatchXplain's matching engine to process two-sided market problems, and MatchXplain's mechanism selection and comparison system. Each of these components adopt a multi-layered and scaffolded structure when presenting the results.

3.1 Preference Analysis

Preference mapping and analysis is critical to understanding the dynamics between agents in elections [Szufa *et al.*, 2020] and matching markets [Boehmer *et al.*, 2024]. MatchXplain introduces a comprehensive framework designed to analyze input preference profiles across three dimensions: popularity, similarity, and ties. This approach employs established algorithms and visualization techniques to illustrate the dynamics of two-sided matching markets.

Popularity Analysis

To quantitatively describe the desirability of agents in a given preference profile, MatchXplain utilizes the Borda count method, a well-studied positional voting mechanism that aggregates individual rankings into a collective preference order [Saari, 2023]. This method is well-suited for extracting numerical scores from ranked preferences, which are then used to determine the overall popularity of each agent in the market.

Borda Count. MatchXplain adapts the traditional Borda Count metric, typically used for complete and strict preferences, to handle ties and incomplete preferences. This is done through Averaged and Pessimistic Borda adjustments [Terzopoulou and Endriss, 2021]. The Averaged Borda approach calculates scores for incomplete preferences by averaging points for unranked agents, effectively assuming they are tied in rank. The Pessimistic Borda approach assigns the lowest possible scores to unranked alternatives, assuming these agents are strictly unwanted.

Similarity Analysis

To evaluate the similarity between agents' preferences, MatchXplain employs the Kendall tau distance, another well-studied metric to compute statistical correlation [Kendall, 1938].

Kendall tau distance. The Kendall tau distance quantifies the similarity between two sets of rankings by identifying and counting pairs that are in a different order between the two rankings (these are called discordant pairs) [Kendall, 1938]. To handle weak and/or incomplete preferences, MatchXplain considers unranked agents as being tied at the bottom of preference lists, and ties between pairs of agents are counted as half the value of a discordant pair. The platform additionally integrates *principal component analysis* (PCA) [Jolliffe, 1986] and *hierarchical clustering* [Johnson, 1967] to describe how agent preferences are clustered.

Tie Analysis

To identify and classify ties in preference profiles, MatchXplain employs two mechanisms: the *Louvain* algorithm for community detection, and the *Tie Ratio* metric developed specifically for MatchXplain to quantify and visualize the extent of ties.

Louvain Algorithm. The Louvain algorithm [Blondel *et al.*, 2008] detects community structures in large networks. MatchXplain applies this by representing preference ties as an adjacency matrix, which is used by the algorithm to cluster agents based on shared ties. This approach groups agents with similar desirability, reflecting their relative appeal.

Tie Ratio Method. MatchXplain additionally quantifies the extent of ties using a newly-developed *tie ratio* method. For each agent $a_i \in A$, the *individual tie score* S_i, is calculated as $S_i = A_{\text{ties},i} - T_i$, where $A_{\text{ties},i}$ is the sum of alternatives in a_i's preference list that are involved in ties (possibly different ties), and T_i is the number of ties in a_i's preference list. The *total tie score* is the summation of individual tie scores across all agents, $S_{\text{total}} = \sum_{i=1}^{n+m} S_i$, where $m = |A|$ and $n = |B|$. The maximum possible individual tie score is $m-1$ for agents in A ($n-1$ for agents in B), occurring when all agents in a preference list are tied, leading to a maximum possible total tie score of $m(n-1)+n(m-1)$. The *tie ratio*, R, is then defined as $R = \frac{S_{\text{total}}}{m(n-1)+n(m-1)}$, providing a normalized measure of the prevalence of ties within a preference profile.

3.2 Matching Engine

The platform's matching engine currently employs five algorithms (with more under development) to compute solutions based on preference data. These solutions satisfy a range of economic properties.

Stable Matching

When the goal is to achieve stability, the platform selects different algorithms based on the nature of the preference data (e.g., strict or weak, complete or incomplete preferences). Each algorithm satisfies a specific variation of stability and addresses different aspects of efficiency, such as the size of the matching

Deferred Acceptance. The most seminal mechanism—proposed by Gale and Shapley [Gale and Shapley, 1962]—is the *deferred acceptance* (DA) algorithm. MatchXplain's stable marriage algorithm is an implementation of the algorithms developed for weak preferences (those with ties) [Irving, 1994]. In the presence of ties, different variants of stability—*weak*, *strong*, and *super* stability—are computed using their respective algorithms.

Stable Marriage with Incomplete Preferences MatchXplain also generates solutions for incomplete preference profiles. Because the problem of finding a maximum size stable solution under incomplete preferences containing ties is APX-Hard [McDermid, 2009], MatchXplain employs an algorithm developed by Kiràly to approximate the maximum size stable solution [Király, 2013]. The implementation achieves 3/2 approximation in this setting. Kiràly further introduces an approximation algorithm to find stable one-to-many solutions with weak and incomplete preferences [Király, 2013].

Top Trading Cycles

To find a Pareto optimal solution (for one side), MatchXplain implements the Top Trading Cycles (TTC) algorithm [Shapley and Scarf, 1974]. The resulting matching is guaranteed to be Pareto optimal but may not be stable [Shapley and Scarf, 1974]. Since TTC is a one-sided allocation mechanism, only one side of agents' preferences are considered, and an efficient allocation is found based on these preferences. This algorithm is only applied when preference profiles are complete. Fixed tie-breaking strategies can the applied when working with weak preference profiles [Ehlers, 2014].

3.3 Mechanism Selection and Comparison

Matchxplain compares potential matchings generated by the matching engine by displaying key properties (such as matching size, number of blocking pairs, and number of envious agents) on *radar charts* for easy interpretation of trade-offs between the solutions. The use of interactive bipartite graphs further presents individual matchings by displaying matched pairs, blocking pairs, exchange cycles, and even envy between agents in a multi-layered manner. These interactive graphs enable the visual exploration and comparison of the recommended solutions.

4 Implementation Details and Data Format

MatchXplain is a web application developed with React, with a backend REST API developed with Flask and Python. All algorithms described in Sections 3.1 & 3.2 are implemented in Python. Users upload **preference data** to MatchXplain as plain text. These preference profiles follow a similar format to that of the *Preflib* preference library [Mattei and Walsh, 2013]. MatchXplain currently accepts four types of two-sided ordinal preference profiles: **SOC**: Complete strict preferences without ties; **TOC**: Complete preferences with permissible ties; **SOI**: Incomplete strict preferences without ties; **TOI**: Incomplete preferences with permissible ties.

5 Concluding Remarks

MatchXplain offers detailed preference analysis and diverse matching algorithms for a variety of objectives and preference types, presenting information in a scaffolded, digestible manner. The components such as the matching engine are currently focused on two-sided matching markets. Yet, its preference analysis tools and layered recommendations can be utilized within other relevant computational social choice problems (e.g. voting theory [Szufa *et al.*, 2020], hedonic games [Kerkmann *et al.*, 2020], fair division [Igarashi *et al.*, 2024], etc.) that deal with ordinal preference data.

Future work includes integrating additional algorithms (one-to-many matchings for weak and incomplete preferences [Irving, 1994; Király, 2013] and a TTC algorithm for weak preferences [Ehlers, 2014]), adding abstraction layers for deeper analysis, and introducing meta-analysis techniques to allow users to adjust or constrain the output to modify matching outcomes. Additionally, there are future plans to enable the platform to suggest modifications to the initially input preferences. Suggestions could include modifying ties in preference profiles, or modifying matching quotas in many-to-one matchings. These problems pose computational challenges.

Acknowledgments

This research was supported in part by NSF Awards IIS-2144413 and IIS-2107173.

References

[Abdulkadiroğlu *et al.*, 2005a] Atila Abdulkadiroğlu, Parag A Pathak, and Alvin E Roth. The New York City High School Match. *American Economic Review*, 95(2):364–367, 2005.

[Abdulkadiroğlu *et al.*, 2005b] Atila Abdulkadiroğlu, Parag A Pathak, Alvin E Roth, and Tayfun Sönmez. The Boston Public School Match. *American Economic Review*, 95(2):368–371, 2005.

[Abdulkadiroğlu *et al.*, 2020] Atila Abdulkadiroğlu, Yeon-Koo Che, Parag A Pathak, Alvin E Roth, and Olivier Tercieux. Efficiency, justified envy, and incentives in priority-based matching. *American Economic Review: Insights*, 2(4):425–442, 2020.

[Banerjee and Johari, 2019] Siddhartha Banerjee and Ramesh Johari. *Ride Sharing*, pages 73–97. Springer International Publishing, Cham, 2019.

[Blondel *et al.*, 2008] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[Boehmer *et al.*, 2024] Niclas Boehmer, Klaus Heeger, and Stanisław Szufa. A map of diverse synthetic stable matching instances. *Journal of Artificial Intelligence Research*, 79:1113–1166, 2024.

[Ehlers, 2014] Lars Ehlers. Top trading with fixed tie-breaking in markets with indivisible goods. *Journal of Economic Theory*, 151:64–87, 2014.

[Eskandanian and Mobasher, 2020] Farzad Eskandanian and Bamshad Mobasher. Using stable matching to optimize the balance between accuracy and diversity in recommendation. In *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization*, pages 71–79, 2020.

[Ferris and Hosseini, 2020] James Ferris and Hadi Hosseini. Matchu: An interactive matching platform. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13606–13607, 2020.

[Flanigan *et al.*, 2021] Bailey Flanigan, Paul Gölz, Anupam Gupta, Brett Hennig, and Ariel D Procaccia. Fair algorithms for selecting citizens' assemblies. *Nature*, 596(7873):548–552, 2021.

[Gale and Shapley, 1962] David Gale and Lloyd S Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.

[Gerding *et al.*, 2013] Enrico H. Gerding, Sebastian Stein, Valentin Robu, Dengji Zhao, and Nicholas R. Jennings. Two-sided online markets for electric vehicle charging. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems*, AAMAS '13, page 989–996. International Foundation for Autonomous Agents and Multiagent Systems, 2013.

[Glitzner and Manlove, 2024] Frederik Glitzner and David Manlove. Matwa: A web toolkit for matching under preferences. *arXiv preprint arXiv:2409.04402*, 2024.

[Goldman and Procaccia, 2015] Jonathan Goldman and Ariel D Procaccia. Spliddit: Unleashing fair division algorithms. *ACM SIGecom Exchanges*, 13(2):41–46, 2015.

[Gusfield, 1987] Dan Gusfield. Three fast algorithms for four problems in stable marriage. *SIAM Journal on Computing*, 16(1):111–128, 1987.

[Igarashi *et al.*, 2024] Ayumi Igarashi, Yasushi Kawase, Warut Suksompong, and Hanna Sumita. Fair division with two-sided preferences. *Games and Economic Behavior*, 147:268–287, 2024.

[Irving, 1994] Robert W Irving. Stable marriage and indifference. *Discrete Applied Mathematics*, 48(3):261–272, 1994.

[Johnson, 1967] Stephen C Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967.

[Jolliffe, 1986] Ian T Jolliffe. Principal component analysis and factor analysis. *Principal component analysis*, pages 115–128, 1986.

[Kendall, 1938] Maurice G Kendall. A new measure of rank correlation. *Biometrika*, 30(1-2):81–93, 1938.

[Kerkmann *et al.*, 2020] Anna Maria Kerkmann, Jérôme Lang, Anja Rey, Jörg Rothe, Hilmar Schadrack, and Lena Schend. Hedonic games with ordinal preferences and thresholds. *Journal of Artificial Intelligence Research*, 67:705–756, 2020.

[Király, 2013] Zoltán Király. Linear time local approximation algorithm for maximum stable marriage. *Algorithms*, 6(3):471–484, 2013.

[Manlove *et al.*, 2002] David F Manlove, Robert W Irving, Kazuo Iwama, Shuichi Miyazaki, and Yasufumi Morita. Hard variants of stable marriage. *Theoretical Computer Science*, 276(1-2):261–279, 2002.

[Mattei and Walsh, 2013] Nicholas Mattei and Toby Walsh. Preflib: A library for preferences http://www. preflib. org. In *Algorithmic Decision Theory: Third International Conference, ADT 2013, Bruxelles, Belgium, November 12-14, 2013, Proceedings 3*, pages 259–270. Springer, 2013.

[McDermid, 2009] Eric McDermid. A 3/2-approximation algorithm for general stable marriage. In Susanne Albers, Alberto Marchetti-Spaccamela, Yossi Matias, Sotiris Nikoletseas, and Wolfgang Thomas, editors, *Automata, Languages and Programming*, pages 689–700, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[Roth and Peranson, 1999] Alvin E Roth and Elliott Peranson. The Redesign of the Matching Market for American Physicians: Some Engineering Aspects of Economic Design. *American Economic Review*, 89(4):748–780, 1999.

[Saari, 2023] Donald G Saari. Selecting a voting method: the case for the borda count. *Constitutional Political Economy*, 34(3):357–366, 2023.

[Shapley and Scarf, 1974] Lloyd Shapley and Herbert Scarf. On cores and indivisibility. *Journal of mathematical economics*, 1(1):23–37, 1974.

[Szufa *et al.*, 2020] Stanisław Szufa, Piotr Faliszewski, Piotr Skowron, Arkadii Slinko, and Nimrod Talmon. Drawing a map of elections in the space of statistical cultures. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems*, pages 1341–1349, 2020.

[Terzopoulou and Endriss, 2021] Zoi Terzopoulou and Ulle Endriss. The borda class: An axiomatic study of the borda rule on top-truncated preferences. *Journal of Mathematical Economics*, 92:31–40, 2021.