MatchU: An Interactive Matching Platform

James Ferris, Hadi Hosseini
Department of Computer Science
Rochester Institute of Technology
{jef1771, hhvcs}@rit.edu

Abstract

MatchU is a web-based platform that offers an interactive framework to find how to form mutually-beneficial relationships, decide how to distribute resources, or resolve conflicts through a suite of matching algorithms rooted in economics and artificial intelligence. In this paper, we discuss MatchU's vision, solutions, and future directions.

Introduction

Matching problems are present in a myriad of real-world situations that concern mutually beneficial decisions for multiple agents. These problems often arise when assigning workers to employers in labor markets, students to colleges, courses to students, nurses to doctors, inheritance to heirs, or other conflict resolutions over scarce resources.
MatchU (MatchU.ai) offers a web-based framework for interactive representation and analysis of a variety of one-sided, two-sided, and probabilistic matching mechanisms. Its primary objective is to foster the adoption of matching algorithms in everyday situations that involve distributing a set of resources or forming beneficial relationships through a suite of algorithms and techniques that are rooted in economics, computer science, and artificial intelligence. From a pedagogical perspective, MatchU leverages interactive design to inspire learners and to facilitate education in a novel animated and customizable framework through *instructional scaffolding* and *inquiry-based* learning. In the following sections, we describe a generic matching model and our desirable properties, and provide a description of matching solutions currently offered by MatchU (see Figure 1). We discuss the steps of generating or modifying instances, manipulating the interface, interacting with algorithms, and utilizing the available tools to make matching decisions.

Matching Model

A matching problem is concerned with finding a mutually beneficial matching between two disjoint sets. Let $N = \{1, \ldots, n\}$ and $M = \{1, \ldots, m\}$ denote two finite and disjoint sets. Let $\succ = (\succ_N, \succ_M)$ be a *preference profile* such
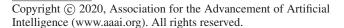
Figure 1: A screenshot of the MatchU platform.

that $\succ_N$ and $\succ_M$ are the preference lists for $i \in N$ and $j \in M$, respectively. Each agent has an associated list $\succ_i$ representing her strict linear order over the members of the opposite set. Thus, agent $i \in N$ prefers $j \in M$ to $j' \in M$ if $j \succ_i j'$, and similarly agent $j \in M$ prefers $i \in N$ to $i' \in N$ if $i \succ_j i'$. A matching problem is *one-sided* when agents in one set, say M, do not have preferences over N. Agents in N may be initially endowed with a member in M. We denote by ω the *endowment profile*, i.e. $\omega = (\omega_1, \ldots, \omega_n)$ such that $\forall i, i' \in N$ and $i \neq i'$ we have $\omega_i \cap \omega_{i'} = \emptyset$. A *matching* $\mu \in \mathcal{M}$ is a mapping between the two sets that indicates an outcome for each agent i as $\mu(i)$. A *matching mechanism* takes as input preference and endowment profiles (which could be empty) and outputs a matching $\mu \in \mathcal{M}$. A mechanism is *probabilistic* if it outputs a probability distribution over a set of deterministic matchings, that is, $\bar{\mu} \in \Delta(\mathcal{M})$.

Desirable Guarantees. A matching μ is **stable** if there exists no pair of agents $(i, j) \in N \times M$ such that they both prefer each other to the (subset of) matching specified by μ. A matching is **Pareto efficient** if no other matching exists that can make at least one agent better off without making any other agent worse off. A matching is **individually rational** if every agent is guaranteed to receive no worse than her initial endowment, i.e., $\forall i \in N$, either $\mu(i) \succ_i \omega_i$ or $\mu(i) = \omega_i$. We focus on **fairness** in a probabilistic sense. A matching satisfies *equal treatment of equals* if it assigns the same chances to agents with identical preferences. A matching is *envy-free* if no agent prefers the lottery assigned to another agent. A mechanism is truthful (or strategyproof) if no agent can benefit from misreporting her preferences.

Matching algorithms

MatchU offers a suite of matching algorithms for two-sided matching, one-sided matching, and probabilistic matching:

Deferred Acceptance (DA). The Deferred Acceptance algorithm proposed by (Gale and Shapley 1962) guarantees a stable solution for two-sided matching problems: Given a preference profile, DA finds a stable matching that is optimal for the proposing side.

Resident Matching (RM). Resident matching is an extension to the stable marriage problem where agents on one side (residents) can only be matched to one hospital on the other side with possibly multiple quotas. The solution is an extension of DA and guarantees a stable matching that is optimal for residents (proposing side) (Roth and Peranson 1999).

Serial Dictatorship (SD). Serial Dictatorship provides a (Pareto) efficient and truthful solution to the one-sided matching problem (Svensson 1999). Given a *priority ordering* of agents, indicating seniority or age, $\sigma = (\sigma_1, \ldots, \sigma_n)$, agent $\sigma_1 \in N$ picks an item from M, $\sigma_2 \in N$ picks an item from the remaining items in M, and so on.

Random Priority (RP). Random Priority (aka Random Serial Dictatorship) is a fair extension of SD when there is no predefined ordering of agents. RP achieves fairness by permuting over all possible orderings of agents and then selecting one uniformly at random. The outcome of RP is a probabilistic matching that assigns to each agent a 'lottery' indicating the agent's chance of receiving each item. RP is Pareto efficient, truthful, and satisfies *equal treatment of equals* (Abdulkadiroğlu and Sönmez 1998).

Probabilistic Serial (PS). Probabilistic serial rule is a well-studied mechanism that simulates a *simultaneous eating* algorithm. Suppose items in M are different types of pizzas. Each agent starts "eating" her top choice pizza at the same rate as every other agent. Once a pizza is consumed, agents move to their next preferred pizza. Similar to RP, the final fractional outcome form a 'lottery' for agents. PS guarantees that the probabilistic matching (lottery) is efficient and envy-free (Bogomolnaia and Moulin 2001). PS can be extended to multiple assignments where agents may receive more than one item (Hosseini, Larson, and Cohen 2018).

Top Trading Cycle (TTC). When agents in N are initially endowed with items from M, the Top Trading Cycle algorithm guarantees a mutually-beneficial exchange between the agents such that the final outcome is Pareto efficient, individually rational, and truthful (Shapley and Scarf 1974). In each step, agents point at their most preferred item that is available. At least one cycle will form, and agents exchange their houses along this cycle and leave. The process continues until no agent is left.

Interactive Framework

Preference and instance generation. MatchU enables users to select an instance by modifying the number of agents, preferences, endowments and quotas (where applicable), and the procedure of the algorithm. MatchU offers two methods for instance generation. An instance of a problem may include a preference profile, endowment profile, and quota constraints. Users either manually specify an instance or they can generate random configurations (i.e. preference profiles and endowments) based on a uniform distribution over all possible configurations. If an instance is sampled randomly, users are able to modify the preferences, endowments, etc. through the interactive components.

Pedagogy and Interactivity. Each mechanism is accompanied with 'Learn' and 'Interact' components. The former, provides a simple description of problems to be tackled and the theoretical guarantees, along with the description of how we solve the problem. The latter, provides tools for generating instances and for step-wise execution of each algorithm. Each algorithm is equipped with rewind, pause/play, and fast-forward controls so that users can go through the steps at their own pace and fully digest how the algorithm works. These steps enable users to adopt and employ the solutions with a reasonable understanding of the mechanisms.

Under the Hood. Matchu is built using Flask for Python. For parts of user interface and our navigation bar and button designs, we utilize CSS frameworks that enable us to partition page layouts and provide additional appealing component designs. Animation tools are developed based on a mixture of CSS transitions and animations, javascript, and in some cases D3.js. Currently, all computation and hosting utilize AWS cloud computing.

Future Plans

We envision two equally important and parallel directions for future work. First, we plan to expand the set of algorithms to include additional solutions and leverage a plethora of sound algorithmic solutions developed over decades and facilitate learning and comprehension of relevant concepts and solutions. Second, we aim at releasing an API through MatchU that enables users to run large-scale executions.

Acknowledgments

References

Abdulkadiroğlu, A., and Sönmez, T. 1998. Random serial dictatorship and the core from random endowments in house allocation problems. *Econometrica* 66(3):689.

Bogomolnaia, A., and Moulin, H. 2001. A new solution to the random assignment problem. *Journal of Economic theory* 100(2):295–328.

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

Hosseini, H.; Larson, K.; and Cohen, R. 2018. Investigating the characteristics of one-sided matching mechanisms under various preferences and risk attitudes. *Autonomous Agents and Multi-Agent Systems* 32(4):534–567.

Roth, A. E., and Peranson, E. 1999. The redesign of the matching market for american physicians: Some engineering aspects of economic design. *American Economic Review* 89(4):748–780.

Shapley, L., and Scarf, H. 1974. On cores and indivisibility. *Journal of Mathematical Economics* 1(1):23–37.

Svensson, L.-G. 1999. Strategy-proof allocation of indivisible goods. *Social Choice and Welfare* 16(4):557–567.