MATWA: A Web Toolkit for Matching Under Preferences

Frederik Glitzner, David Manlove

School of Computing Science, University of Glasgow, Glasgow G12 8QQ, UK
f.glitzner.1@research.gla.ac.uk, david.manlove@glasgow.ac.uk

Abstract

Matching markets, in which agents are assigned to one another based on preferences and capacity constraints, are pervasive in various domains. This paper introduces *MATWA* (**https://matwa.optimalmatching.com**), a web application that offers the most comprehensive collection to date of algorithms for fundamental matching under preference problem classes. *MATWA* provides results of algorithm executions and visualisations of structural properties. It is intended to be a resource for the community of researchers, educators and practitioners, supporting experimentation, as well as aiding the understanding of matching algorithms.

Introduction

Matching Markets Matching markets occur in problems that involve assigning agents to one another, subject to various criteria. Examples include assigning doctors to hospitals, pupils to schools and kidney patients to donors. We focus on the case where the agents (or a subset of them) have *ordinal* preferences over all (or a subset of) the other agents. Such markets can be very large in practice; for example, the National Resident Matching Program assigned nearly 40,000 junior doctors in 2024 (NRMP 2024). Often, many solution concepts can be applied within matching markets, and it can be useful to compare and contrast, and indeed visualize different outcomes.

Contribution and Significance In this paper, we introduce *MATWA* (Matching Algorithm Toolkit Web Application, **https://matwa.optimalmatching.com**), a web application that makes available implementations of over 40 algorithms for the fundamental matching problem classes HOSPITALS / RESIDENTS (HR), STABLE MARRIAGE (SM), CAPACITATED HOUSE ALLOCATION (CHA), STABLE ROOMMATES (SR) and STUDENT-PROJECT ALLOCATION (SPA). For many of these algorithms, implementations are not publicly available or are difficult to use. *MATWA* overcomes this barrier and offers the possibility of easily and quickly testing and comparing different algorithms (see Figure 1 for a screenshot). Overall, this is by far the most comprehensive collection of algorithms for matching under preference problems to date, made accessible through a simple user-friendly online interface.

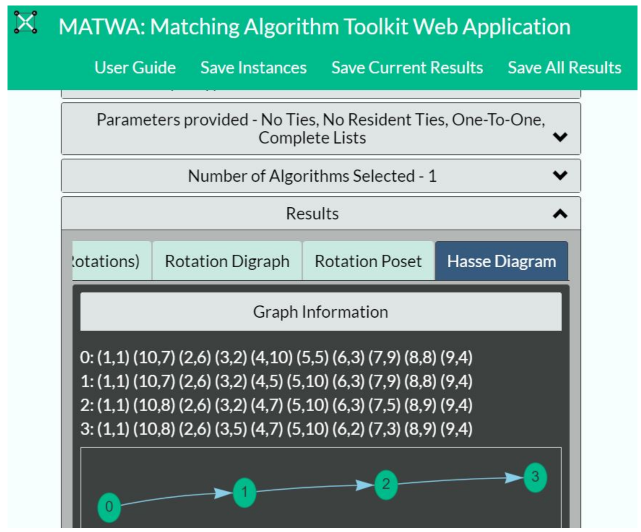

Figure 1: *MATWA*-generated Hasse diagram of SM instance

Background

Preliminaries and Objectives In general terms, matching markets can be characterised by sets of agents, preference lists and capacity restrictions, and the aim is to form a *matching* of the agents that satisfies these constraints and optimises some objective. Many different objectives have been studied (Manlove 2013); for example, it is often useful in practice to maximise the matching size. Furthermore, *stability*, *Pareto-optimality* and *popularity* are common objectives. The fairness of matchings can be measured using the *profile* (a vector whose kth entry gives the number of agents with their kth choice), *regret* (worst rank assigned) and *egalitarian cost* (weighted sum of the profile).

Problem Classes HR (Gale and Shapley 1962) arises, for example, when allocating junior doctors to hospitals. An instance I of HR involves two disjoint sets of agents– *residents* and *hospitals*, where each resident ranks a subset of hospitals in order of preference and vice versa. Each resident has capacity 1 and each hospital has a positive integral capacity.

SM (Gale and Shapley 1962) is a restriction of HR where the capacity of every agent is 1. It arises, for example, when allocating people in a mentoring scheme. CHA (Shapley and Scarf 1974) is also a variant of HR where hospitals do not have preferences over the residents. It arises, for example, when allocating a set of indivisible goods among a set of applicants. SR (Gusfield and Irving 1989) is a generalisation of SM where there is no bipartition of the agents. It arises, for example, when allocating players in sports tournaments. SPA (Abraham, Irving, and Manlove 2007) is a generalisation of HR where agents contain *students*, *projects* and *lecturers*, and students have preferences (usually over projects). It arises, for example, when allocating final-year projects. SPA-S is a variant of SPA in which lecturers also have preferences over an acceptable subset of students.

MATWA offers algorithms for each problem class above, focusing mainly on those that produce stable, Pareto optimal or popular matchings that satisfy further optimality criteria such as maximum size or fairness conditions on the profile.

Existing Tools Various software tools have been developed for matching problems. For example, *MatchU*, previously showcased at AAAI (Ferris and Hosseini 2020), and *AlgMatch* (Lau and Ormond 2023) offer a small subset of our matching algorithms and visualise their steps for educational purposes. However, they are not primarily research tools and only allow small instances, no uploads, downloading of results, or comparison of different algorithms. *MATWA* offers all of these features. Of course, custom code or implementations based on frameworks such as NetworkX (Hagberg, Swart, and S Chult 2008) or CP-SAT (Google Inc. 2018) are also possible. There are also code repositories available such as the Python package `Matching` (Wilde, Knight, and Gillard 2020), which offers one stable matching algorithm for each of SM, HR, SR and SPA. However, the number of algorithms is small and it is not possible to compare different algorithms.

MATWA System

Features and Experience Here, we focus on the front-end interface of *MATWA*, but the principles also apply to the API endpoints. From the landing page, the user selects a problem class and chooses the instance type. There are four ways to provide data: a text box, an upload mechanism, benchmark instances and a random generator. Up to 14 parameters can be specified for the random generator, such as the number of agents, their capacities and distribution, the lengths of preference lists and the probability of ties within them.

After submitting the input, the system responds with a list of applicable algorithms that the user can select from. Then, the results are provided in individual tabs, each typically showing a matching as a set of pairs and as a highlighted set of entries in the preference lists. The statistics and further visualisations shown depend on the problem class and algorithm, but can include the cardinality, egalitarian cost and profile. Some of the other visualisations available are *switching graphs* for CHA, *rotation posets* for SR and SM, and *rotation digraphs* and *Hasse diagrams* for SM. The output can also be exported; for example, random instances can be downloaded as text files for replicability, and matchings and visualisations can be saved as interactive html files.

An up-to-date list of our algorithms is available in the MATWA User Guide (2024). Many of these do not, to the best of our knowledge, have publicly available implementations, such as the algorithm by Tan and Hsueh (1995) to find a *stable partition* or the algorithm for generating the switching graphs in CHA (McDermid and Irving 2008).

Research and Educational Applications *MATWA* is intended for researchers, students and practitioners alike. It can be used to obtain an overview of structural properties of a problem instance, through the computation of stable partitions, or by visualising the Hasse diagram and the rotation poset. It can also be used to demonstrate fairness in different matching mechanisms by applying different algorithms to the same problem instance (or multiple instances) and comparing the size, regret and profile of their results (or summary results). For example, Glitzner (2023) used *MATWA* to evaluate five different algorithms for the SPA and SPA-S problem classes on randomly generated problem instances. The experiments were intended to study the usefulness of the algorithms in practice and, specifically, investigated how the matching sizes, egalitarian costs per agent group, and computation times varied across the algorithms as either the size of randomly generated instances grew or the agent popularity increased. The experimental results suggested that, as the project popularity factor increases, the matching size decreases but starts to stabilise after a certain threshold.

Implementation and Evaluation *MATWA* was developed by at least 20 contributors over a period of 24 years and consists of independent front- and back-end services, which are both hosted on a University of Glasgow server. The front-end uses bootstrap, JavaScript and jQuery. The back-end is implemented as a Spring Boot REST API in Java. There are testing procedures in place and Lazarov (2018) and Glitzner (2023) conducted user studies on versions of the system with participants having different levels of prior experience with matching problems, and established that the system is highly intuitive, usable and useful for the suggested tasks.

Conclusion and Future Work

In conclusion, *MATWA* is a comprehensive web application that offers more than 40 algorithms for various fundamental matching under preference problem classes. It is a valuable resource for research and teaching through its user-friendly front-end interface and openly exposed back-end API. *MATWA* is especially useful for facilitating easy testing, comparison and analysis of different matching algorithms, which has been demonstrated by past research projects and usability studies performed on the system. In the future, we plan to integrate more algorithms and visualisations and extend the multi-instance support.

For an extended version of this paper that includes, for example, more formal definitions of the problem classes and a diagram indicating their relationships, a comparative summary of existing software, and more details on the usage and evaluation of *MATWA* and its API endpoints, see Glitzner and Manlove (2024).

Acknowledgements

MATWA was a collaborative effort and the full list of people who contributed can be found at **https://matwa.optimalmatching.com/matching/manual**. Frederik Glitzner is supported by a Minerva Scholarship from the School of Computing Science, University of Glasgow. David Manlove is supported by the EPSRC, grant number EP/X013618/1. We would also like to thank the anonymous AAAI reviewers for their positive feedback and helpful suggestions.

References

Abraham, D.; Irving, R.; and Manlove, D. 2007. Two algorithms for the Student-Project Allocation problem. *Journal of Discrete Algorithms*, 5(1): 73–90.

Ferris, J.; and Hosseini, H. 2020. MatchU: An Interactive Matching Platform. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(09): 13606–13607.

Gale, D.; and Shapley, L. 1962. College admissions and the stability of marriage. *American Mathematical Monthly*, 69: 9–15.

Glitzner, F. 2023. *Student-Project Allocation in the Matching Algorithm Toolkit*. BSc Computing Science Level 4 Project Dissertation, University of Glasgow. https://glitznerf.github.io/files/level4project.pdf (accessed 2024-11-29).

Glitzner, F.; and Manlove, D. 2024. MATWA: A Web Toolkit for Matching under Preferences. arXiv:2409.04402.

Google Inc. 2018. CP-SAT Solver. https://developers.google.com/optimization/cp/cp_solver (accessed 2024-11-29).

Gusfield, D.; and Irving, R. 1989. *The Stable Marriage Problem: Structure and Algorithms*. MIT Press.

Hagberg, A.; Swart, P.; and S Chult, D. 2008. Exploring network structure, dynamics, and function using NetworkX. Technical report, Los Alamos National Lab (LANL), Los Alamos, NM (United States).

Lau, L.; and Ormond, C. 2023. AlgMatch. https://callumormond.github.io/individual-project/ (accessed 2024-11-29).

Lazarov, B. 2018. *A web app for visualising matching algorithms*. BSc Computing Science Level 4 Project Dissertation, University of Glasgow.

Manlove, D. 2013. *Algorithmics of Matching Under Preferences*, volume 2 of *Series on Theoretical Computer Science*. World Scientific. ISBN 978-981-4425-24-7.

MATWA User Guide. 2024. MATWA User Guide. https://matwa.optimalmatching.com/matching/manual (accessed 2024-11-29).

McDermid, E.; and Irving, R. 2008. Popular Matchings: Structure and Algorithms. Technical Report TR-2008-292, University of Glasgow, Department of Computing Science.

NRMP. 2024. 2024 Main Residency Match by the Numbers. https://www.nrmp.org/match-data/2024/06/results-and-data-2024-main-residency-match/ (accessed 2024-11-29).

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

Tan, J. J.; and Hsueh, Y.-C. 1995. A generalization of the stable matching problem. *Discrete Applied Mathematics*, 59: 87–102.

Wilde, H.; Knight, V.; and Gillard, J. 2020. Matching: A Python library for solving matching games. *Journal of Open Source Software*, 5(48): 2169.