Strategy-Proof Neural Network-Based Mechanism for Two-Sided Matching

Tomohiro Shimamura, Zhaohong Sun, & Makoto Yokoo

Kyushu University

March 2025

Table of Contents

Motivation

- Standard two-sided matching model assumes that each college has a strict maximum quota / capacity limit.
- It is often the case that the capacity has some flexibility.
- Furthermore, the capacity can be extended with some additional costs (e.g., hiring part-time lecturers), assuming by doing so, we can (significantly) increase students' welfare.
- A strategy-proof mechanism that can handle such flexible and expandable quotas (under the trade-off between students' welfare) is not explored well so far.
- Let neural networks find a good mechanism!

Possible Application: Graduate Admissions at Kyushu University

- Graduate admissions process for master's students, where students are assigned to research labs.
- Each lab has a target recruitment quota, but it can be flexible to some extent.
- A preference of a lab is not well-defined (different types of administration exams, different types of students), so fairness is not a strict requirement.
- Students' welfare is important to keep their motivation.

Table of Contents

Model

- S: the set of students, where $|S| = n$
- C: the set of colleges, where $|C| = m$
- Each $s \in S$ (or $c \in C$) has a strict preference over $C \cup \{\emptyset\}$ (or $S \cup \{\emptyset\}$)
- Each college c has its (soft) capacity limit q_c
- $\widehat{q}$ is the amount of extra seats that can be assigned to some colleges.
- Let Y represent a matching, where Y_s is the college where s is assigned, and Y_c is the set of students assigned to c.

Capacity violation / welfare

- The cost / loss function related to capacity violation is given by $f(\nu(Y))$, where $\nu(Y) = \sum_c \max(0, |Y_c| - q_c) - \hat{q})$, $f(x) = 0$ for $x \leq 0$, and f is an increasing function.
- We assume students' welfare is measured by the total Borda score: if student s is matched with her k-th choice in Y, her Borda score $B_s(Y) = m + 1 - k$. The total Borda score $B(Y) = \sum_s B_s(Y)$.

Desirable properties

fairness: No student has justified envy; student s has justified envy toward another student s' is s prefers college c over her assignment, while c prefers s over s'.

Pareto efficiency: cannot improve any student group without hurting other students

Nonwastefulness: cannot improve any single student without hurting other students

strategy-proofness: no student obtains a better match by misreporting her preference.

Table of Contents

Serial Dictatorship

- A common ordering among students (Master List, ML) exists.
- Assign students sequentially according to ML; a student can choose her most preferred college as long as there exists an available seat (including an extra seat).
- strategy-proof, Pareto efficient, strictly satisfy capacity constraints (the loss related to the capacity is guaranteed to be zero), not fair

Artificial Cap DA

- Allocate extra seats to some colleges in some way (independently from students' preferences)
- Run standard DA
- strategy-proof, fair, strictly satisfy capacity constraints, can be wasteful

Flexible DA [?]

- A round-robin ordering among colleges is given.
- It uses a DA-like procedure, where each college sequentially accepts the remaining most preferred applying student if its seat is available (including an extra seat)
- strategy-proof, fair, strictly satisfy capacity constraints, can be wasteful (but usually better than ACDA since extra seats are allocated based on students' preferences)

Table of Contents

Menu mechanism: overview [?]

- A general method to construct a strategy-proof mechanism
- A set of choices (menu) is presented to each student s, which is created independently from the declared preference of s.
- Based on the declared preference of s, her best choice within the menu is selected.
- It is strategy-proof by definition
- Satisfying feasibility is a challenging part.

Menu mechanism for auction [?]

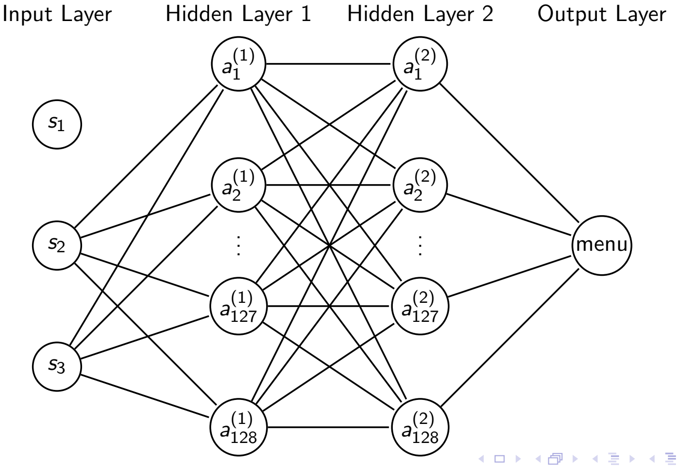
- Allocating a single good to students
- For each student s, her price p_s is presented (which is determined independently from her declared value)
- s can choose either: (i) buy the good paying p_s, or (ii) do not buy the good
- Assuming her value is v_s, then she would buy if $p_s < v_s$, not otherwise.
- If we choose $p_s = \max_{s' \neq s} v_{s'}$, feasibility (at most one student buys the good) is satisfied.

Menu mechanism: in our context

- For each student s, a list of colleges (including $\emptyset$) is presented. The list is determined independently from her preference.
- s chooses the best college from the list.
- It is automatically strategy-proof, but we need to make sure the menus are created such that capacity constraints are satisfied appropriately (considering the trade-off between students' welfare).

NN structure

- Using PyTorch version 2.5.1+cu124
- learning rate: 0.01, optimiser: adam
- we create a network for each student. The loss function is calculated by the aggregated outputs (the combination of students' choices).



Objective / Loss function

Without fairness: $C_1 \cdot f(\nu(Y)) - B(Y)$

With fairness: $C_1 \cdot f(\nu(Y)) - B(Y) + C_2 \cdot Ev(Y)$, where $Ev(Y)$ is the number of student pairs (s, s') where s has justified envy toward s' in Y.

We choose $f(x) = x$, $C_1 = 4$, and $C_2 = 0.5$

Table of Contents

Learning Settings

- 10 students, 6 colleges, the quota of each college is [2,2,1,2,1,2], there is one extra seat.
- The colleges' preferences are randomly generated (and fixed for the overall evaluation)
- 1 epoch contains: generating students' preferences, obtaining matchings, and applying feedback.
- 1,000 epoch for each trial. batch size: 32

Comparison

mechanism	average Borda	envy violation	capacity violation
SD	4.7	1	0.0
ACDA	4.5	0.0	0.0
WO fairness	5.03	18.75	2.85
WT fairness	5.18	18.3	4.1

Conclusions and Future works

- Although the Borda count is good (since quotas are expandable), the learned mechanism is not very good.
- The values of the loss function seem larger than the result of other mechanisms.
- Seems 1000 epochs is not enough?
- Need to try different network structures?
- Need comparison with flexible DA.

Another topic: Automated mechanism design using Constraint Satisfaction Problem (CSP)

- There exists a set of n variables X. Each variable x_i takes its value from a discrete, finite domain D_i
- Constraints are defined as nogoods. A nogood $((x_i, d_i), (x_j, d_j))$ means this combination is not allowed.
- The goal of a CSP is to find the value assignment of all variables that does not include any nogood.
- A Satisfiability Problem (SAT) is one kind of a canonical form of a CSP; any CSP can be converted into SAT automatically.
- Although CSP/SAT is NP-complete in general, there exist many open-source programs (SAT solvers) that can solve really huge SAT instances efficiently in practice (for a large problem instance, the main memory would be a bottle-neck).

Matching mechanism design as CSP

- Fix the number of students/colleges, colleges' preferences, and quotas, etc.
- Enumerate all possible students' preference profiles
- For each profile, enumerate all matchings that satisfy desirable properties (except for strategy-proofness)
- A profile is a variable, and its domain is the possible matchings
- Enumerate all nogoods derived from strategy-proofness
  - Assume θ and θ', which are different only for student s's type.
  - For $x \in D_\theta$ and $x' \in D_{\theta'}$, if s prefers x' over x, then $((\theta, x), (\theta', x'))$ is a nogood.
- If the obtained CSP is unsatisfiable, then no strategyproof mechanism exists.
- If it is satisfiable, the solution is a part of the mechanism (under the fixed colleges' preferences, etc.)

Research Issue

- If the obtained CSP is unsatisfiable, we have an impossibility theorem (though making a human-readable proof is still an issue).
- If the obtained CSP is satisfiable for several different settings, it is very likely a strategy-proof mechanism exists, but we just have many huge tables; we have no intuition how the mechanism looks like.
- Idea: can we use (deep) reinforcement learning to obtain rules that constitute the mechanism?

Applying reinforcement learning

- Assume a state (in the reinforcement learning) is represented as a students' preference profile, a partial matching, and *forbidden pairs*.
- A forbidden pair (s, c) is a pair of student s and college c, which means that s cannot be accepted to college c.
- In a given state, if all students can be accepted to her first-choice college (which is not forbidden), we assume it is a terminal state.
- Otherwise, the mechanism takes an action, which is either (i) a particular pair (s, c) is forbidden, or (ii) a particular pair (s, c) is added to the partial matching; then a new state is obtained.
- By reaching a correct (or wrong) terminal state (which is defined by the CSP solution, some positive (or negative) reward is given.
- After the learning process converges, we know the correct action for each state, which gives us a more detailed description of the mechanism obtained by CSP.

References