Making (Good) Matches: Building and Piloting an Online Platform for Experimental School Choice Research

Drew Beckmen

Advisor: Professor Yusuke Narita, PhD

CSEC 491 Senior Project
Yale University
May 1, 2024

Abstract

In this project, I examine the role of matching as it is applied to the problem of U.S. public school choice. Across the country, families submit ordered preferences to their districts, which function as centralized clearinghouses for the two-sided market over student demand and available seats. Although most districts leverage student-optimal, strategyproof mechanisms such as the deferred acceptance algorithm, many families still submit untruthful preferences in an attempt to "game" the system and obtain a better outcome. My project fits into the broader game theoretic literature on truth-telling behavior in the school choice problem. Using modern web technologies such as NextJS, FastAPI, and MongoDB, I developed MatchEd, a platform for configuring and running online, crowd-sourced experiments related to the school choice problem. The platform offers researchers visual and programmatic interfaces via the web and its REST API respectively, and it generates links that can be inputted into online crowd-sourced marketplaces such as Amazon Mturk.

To test the platform, I conducted a pilot study to examine participants' learning dynamics in repeated school choice games. I analyze the differences in truth-telling between a control group that plays a one-shot game and an experimental group that plays five simulated practice rounds before the actual game. The results support that the dominant strategy is "learnable," even over a small number of games. Given the results of the pilot study, I argue that further research is needed to develop practical methods of countering sub-optimal behavior among participants in strategyproof school choice matching mechanisms.

Contents

1 Introduction **2**
- 1.1 Background and Related Literature . . . 2
- 1.2 Project Significance . . . 6

2 Methods **7**
- 2.1 Building MatchEd . . . 7
- 2.2 Experimental Design for MatchEd Pilot . . . 9

3 Results **11**
- 3.1 General Availability of MatchEd . . . 11
- 3.2 Pilot Study Data . . . 12

4 Future Work **16**

5 Conclusion **17**

6 References **17**

7 Appendix **18**

1 Introduction

1.1 Background and Related Literature

Matching algorithms play a critical role in clearing markets in which price is not the sole determinant of outcomes. From facilitating organ transplants to assigning medical students to their residency programs, matching algorithms underlie many of the unseen markets that pervade our everyday lives.[1] Without sound design, matching markets can unravel as participants withdraw from the system. In 2009, Chicago Public Schools (CPS) experienced a market failure as parents realized that some of the best performing students received poor outcomes from the match. At the time, CPS assigned matches via the immediate acceptance (IA) algorithm, commonly known as "Boston mechanism" for its use in the Boston school system until 2005. CPS abandoned the 2009 process and replaced the IA algorithm with a form of serial dictatorship.[2] As public school choice expanded globally in an effort to reduce segregation and improve student outcomes, market designers intervened to solve the assignment problem of students to schools.

Rise of Strategyproof Algorithms

Gale and Shapley discovered the deferred acceptance algorithm in their seminal paper "College Admissions and the Stability of Marriage" published in 1962. In the paper, they prove two important algorithmic properties: DA is stable, and DA is optimal for participants on the "proposing" side of the market. Gale and Shapley define stable assignments as those without a blocking pair. A blocking pair occurs when participant on either side of the market mutually prefer each other to their current matches. More formally, consider a matching μ that assigns each $s \in S$ to some $u \in U$. $(s, u) \in S \times U$ is a blocking pair if:

$$u \succ_s \mu(s)$$
$$s \succ_u \mu(u)$$

Economists did not formulate school choice as a mechanism design problem until 2003. That year, Abdulkadiroğlu and Sönmez described the generic school choice problem as:

- a set of students: $N = \{1, \ldots, |N|\}$
- a set of schools: $S = \{s_1, \ldots, s_{|S|}\}$
- each school has a capacity q_s
- each student has a strict preference relation P_i over $S \cup \{i\}$ where i represents remaining unmatched.
- each school has a weak ordering $\succsim_s$ over $N \cup \{s\}$

This setting reduces to the Gale-Shapley setting when schools' weak preferences are tie-broken. In their paper, Abdulkadiroğlu and Sönmez discuss the benefits of switching to strategyproof algorithms such as DA or top trading cycles (TTC) compared to the widely used IA algorithm.

[1] For more examples, see Roth (2016). *Who Gets What — and Why: The New Economics of Matchmaking and Market Design*

[2] Similar story lines have unfolded across the country. See "How Game Theory Helped Improve New York City's High School Application Process" (2014) in the *New York Times*.

With a strategyproofness guarantee, families should not have to strategize when submitting preferences to the school district. This dominant strategy incentive compatibility (DSIC) guarantee makes the market less manipulable and limits the inequities between informed and uninformed participants.

Description and Properties of Matching Algorithms

The immediate acceptance (IA) algorithm is the most intuitive matching algorithm, and, as I discuss later, many participants make strategic decisions under the false assumption that their district uses IA. In IA, seats are initially allocated at each school to those students who rank the school first. Next, seats are allocated to those who rank the school second (assuming the school has any remaining capacity), and so on.

The algorithm emphasizes welfare maximization by prioritizing students' first choices, an axiomatic property formalized by Kojima and Ünver in 2014. However, the priority given to high ranks necessitates a sacrifice in strategyproofness. Students can easily manipulate outcomes by ranking schools higher to improve their chances of admission. In their applied experimental work, Chen and Sönmez (2006) observed higher rates of preference manipulation in IA than DA or TTCA. A simple example illustrates the power of preference manipulation to obtain a better outcome:

> S is the set of students with $|S| = 3$. U is the set of schools with $|U| = 3$. The school capacities $q_i = 1\ \forall i$.
>
> Table 1: IA Not Strategyproof
>
> | Student Preferences | School Preferences |
> |---|---|
> | $u_1 \succ_{s_1} u_2 \succ_{s_1} u_3$ | $s_3 \succ_{u_1} s_2 \succ_{u_1} s_1$ |
> | $u_1 \succ_{s_2} u_2 \succ_{s_2} u_3$ | $s_2 \succ_{u_2} s_3 \succ_{u_2} s_1$ |
> | $u_1 \succ_{s_3} u_2 \succ_{s_3} u_3$ | $s_2 \succ_{u_3} s_3 \succ_{u_3} s_1$ |
>
> In the first round, all students apply to their first choice: school u_1. u_1 accepts s_3, so s_1 and s_2 must apply to their second choice: u_2. u_2 accepts s_2, so s_1 is assigned to the last remaining school u_3.
>
> Had s_1 reported the preferences $u_2 \succ_{s_1} u_1 \succ_{s_1} u_3$, they would have been assigned to u_2 in round one, which is preferred to u_3. This simple example illustrates the manipulability of the algorithm, especially when school popularity is well-understood. More specifically, if s_1 recognized that all students wanted u_1 and they were a low priority student, they could have misreported preferences to gain admission to their second favorite school.

Formally, the IA mechanism is defined by the following iterative algorithm. Let S be the set of students, U be the set of schools, M be the set of matchings, and q_i be the capacity of school $u_i \in U$.

Algorithm 1 Immediate Acceptance

repeat

Step 1: Each student $s \in S$ proposes to their first choice school. Each school u_i immediately accepts its q_i top-ranking applicants, rejecting all others. q_i is set to zero for schools with at least q_i applicants, otherwise it's set to the remaining capacity.

Step l: Remaining unmatched students apply to their lth favorite school. Schools accept students up to capacity q_i, rejecting all others. Capacity are updated.

until no rejections are issued

A simple counter-example proves the presence of justified envy in outcomes from IA. The presence of justified envy implies that the algorithm does not produce stable outcomes. Consider the following setting:

> S is the set of students with $|S| = 3$. U is the set of schools with $|U| = 3$. The school capacities $q_i = 1 \ \forall i$.

Table 2: IA Produces Justified Envy

Student Preferences	School Preferences
$u_1 \succ_{s_1} u_3 \succ_{s_1} u_2$	$s_2 \succ_{u_1} s_1 \succ_{u_1} s_3$
$u_1 \succ_{s_2} u_3 \succ_{s_2} u_2$	$s_1 \succ_{u_2} s_2 \succ_{u_2} s_3$
$u_3 \succ_{s_3} u_2 \succ_{s_3} u_1$	$s_1 \succ_{u_3} s_2 \succ_{u_3} s_3$

> In the first round, s_1 and s_2 apply to u_1, and u_1 accepts s_2. s_3 applies to u_3, and u_3 accepts. s_1 ends up matched to the last remaining school u_2.
>
> This assignment produces justified envy since s_1 prefers u_3 to its current assignment (u_2), and u_3 prefers s_1 to its current assignment (s_3).

The deferred acceptance algorithm (DA), which is the most commonly used strategyproof algorithm, functions differently. Unlike IA where matches are permanent once assigned, all matches in DA are tentative until the algorithm terminates. The potential for matching and un-matching during the algorithm's execution makes it less intuitive to market participants.

In the following pseudocode, M is a temporary matching, U is the set of schools, S is the set of students, and $U_s \subseteq U$ is the subset of schools to which student s has proposed. We define a complete matching as one in which every party is matched. Note that the following algorithm assumes that each student is matched to one school and vice versa. The algorithm easily extends to the setting formulated by Abdulkadiroğlu and Sönmez in which each school has a capacity q_i where $i \in \{1, \ldots, |U|\}$: simply add a school i with capacity q_i to the set of schools U q_i times.

Algorithm 2 Deferred Acceptance

```
Initialization:
M ← ∅
U_s ← ∅ for all s ∈ S
repeat
    s ← an arbitrary unmatched student such that U_s ≠ U
    u is favorite school of s in U \ U_s
    if u is unmatched in M then
        match s to u
    else
        if s ≻_u M(u) then
            unmatch u from M(u)
            match s to u
            U_M(u) ← U_M(u) ∪ {u}
        else
            keep matching the same
            U_s ← U_s ∪ {u}
        end if
    end if
until M is a complete matching
```

A proof for the stability property is provided in the appendix. The appendix also includes a description of the top trading cycles algorithm, another notable strategyproof matching algorithm used in some school districts.

Persistent Preference Manipulation

In a February 2023 review paper, Alex Rees-Jones and Ran Shorrer argue that the most pressing work to be done in the field of education market design is by behavioral economists. The idealized rational framework led to significant developments over the past two decades that ameliorated many of the market failures undermining the public school choice processes. However, a growing body of literature grapples with persistent preference manipulation despite the use of strategyproof mechanisms. This sub-optimal behavior by market participants reduces overall welfare and motivates additional experimental and behavioral economic research.

Chen and Sönmez published the first experimental study on school choice in 2006, and their experimental setup has been replicated many times since. The experiment follows an "induced value method" in which each school is tied to a monetary payoff. If participants successfully get matched to their favorite school, they receive the highest possible payoff. The payoffs monotonically decrease with each subsequent school in the preference ordering. Assuming participants attempt to maximize their financial reward, any out-of-order rankings implies suboptimal preference manipulation.

When analyzing the results, the authors quantify the level of preference manipulation in the IA, DA, and TTCA mechanisms. Consistent with theory, the results reveal significant preference manipulation under the IA mechanism. Thus, even though school districts tout the proportion of students receiving their first choice under IA, that "first choice" often does not reflect true student preferences. Notably, preference manipulation persisted in the DA and TTCA conditions as well: between 28 and 57 percent of participants submitted untruthful preferences.

As evidence mounted that strategyproof mechanisms reduced but did not fully eliminate suboptimal behavior, researchers began examining factors that influence truth-telling behavior. In Guillen and Hakimov (2017), the authors showed that providing participants with a detailed description of the algorithm decreased truth-telling compared to simple instructions that defined strategyproofness and the associated dominant strategy. Additionally, Ding and Schotter (2019) experimentally tested the impact of intergenerational advice on truth-telling behavior. The experiment hoped to model the information sharing dynamics among families. The researchers found that social learning in intergenerational sequences of one-shot games negatively impacts participant behavior. However, when individuals play the same game repeatedly, the authors found that participants converged to their dominant strategy through a process of iterative learning. More specifically, the rate of truth-telling monotonically increased among subjects playing the game repeatedly. These findings align with traditional game theory findings on convergence in games with dominant-strategy equilibria[3], and they support the use of experiential learning—as opposed to social learning—to help families understand school choice mechanisms.

1.2 Project Significance

To my knowledge, no online experiments have been conducted in the field of school choice mechanism design. Traditionally, participants are brought into a lab to read instructions, rank their preferences, and learn their results. The MatchEd platform offers researches a flexible tool to conduct large-scale online experiments to assess theories of participant behavior. Researchers can experimentally modify the instructions, payoff profiles, and more to investigate hypotheses related to strategic decision-making in strategyproof mechanisms. Crowd-sourced online experiments are particularly useful for testing behavioral economic hypothesis, such as those described by Rees-Jones and Shorrer (2023). Rees-Jones and Shorrer list many possible theories to explain suboptimal behavior including reliance on improper heuristics (such as assuming DA functions like IA), fear of rejection or loss, and misguided probability perceptions. The source code for MatchEd is open source, and my hope is that others in the community leverage the project for their own research. See the appendix for guidance on how to run and develop MatchEd on your own machine.

In the aforementioned Ding and Schotter paper, participants in the "Repeated" condition played 20 consecutive rounds of the game. Their findings on the benefits of experiential learning are not surprising given they align with the theory of games with dominant strategies. Although Ding and Schotter use the "repeated" case to contrast with the intergenerational condition, which is their main point of emphasis, I focus on learning via repeated games to potentially address the issue of low rates of truth-telling.

I view experiential learning via repeated play as a potentially beneficial tool for families preparing for the school matching process. Ding and Schotter's model of repeated games would not work in practice because it is synchronous and time-consuming. However, a shorter—on the order of five games—online simulation that allows participants to experience the mechanism could push participants toward their dominant strategy when it comes time to submit their actual schooling preferences. The pilot study conducted with MatchEd lays the groundwork for future policy-driven research that seeks to improve the performance (both efficiency and welfare) of school choice mechanisms by reducing the number of participants playing weakly dominated strategies.

[3]See Beggs (2005): "On the convergence of reinforcement learning"

2 Methods

2.1 Building MatchEd

Application Architecture

MatchEd is a full-stack, mobile-friendly web application for conducting online mechanism design experiments. The application allows users to design experiments as school choice problems akin to the design used in Chen and Sönmez (2006). MatchEd follows a standard three-tier application structure with MongoDB at the data layer, FastAPI as the backend, and NextJS as the frontend.

MongoDB, specifically its cloud database-as-a-service offering MongoDB Atlas, fit the requirements of the application perfectly. Its flexible schema allowed for rapid iteration during development, and it also made modeling complex data objects far easier. The MatchEd data model consists of `experiments`, each of which can contain many `conditions`. Each `condition` contains many `schools` and `students`, which are mapped one-to-one to experimental `participants`. The ability to embed complex objects paired with MongoDB's rich aggregation framework reduced the amount of time spent designing the schema and implementing application logic. Lastly, by running MongoDB in the cloud, I avoided the complexity of running a local database server during development.

FastAPI[4] is a Python framework for building performant RESTful APIs. FastAPI is built on top of the ASGI framework Starlette and the data validation library Pydantic. As an asynchronous server gateway interface, Starlette can handle a large number of concurrent connections efficiently, which allows FastAPI to overcome the traditional performance limitations of a Python application. Pydantic allows developers to define models typed with Python type hints, enabling easy schema validation and serialization/de-serialization. By incorporating Pydantic for all data modeling, FastAPI offers built-in data validation normally absent from a dynamically typed language such as Python. The last major benefit of FastAPI is its compatibility with OpenAPI standards. This allows for auto-generated API documentation, which improves usability, platform uptake, and development velocity.

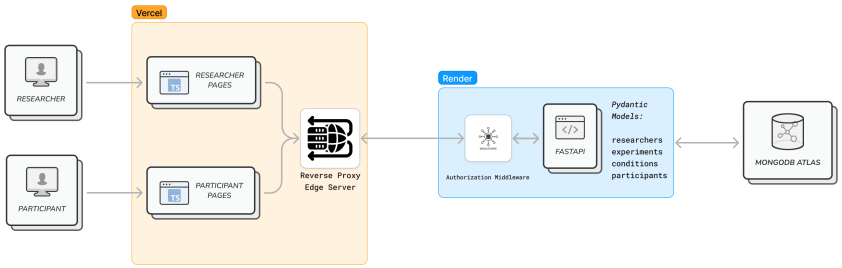


Figure 1: Architecture Diagram

NextJS is a modern web framework built on top of Meta's React library that provides features such as serverless APIs, client-side and server-side rendering, and application middleware. Although NextJS can be used as a full-stack framework, I only leverage its server-side capabilities

[4]To read more about FastAPI, see the documentation here.

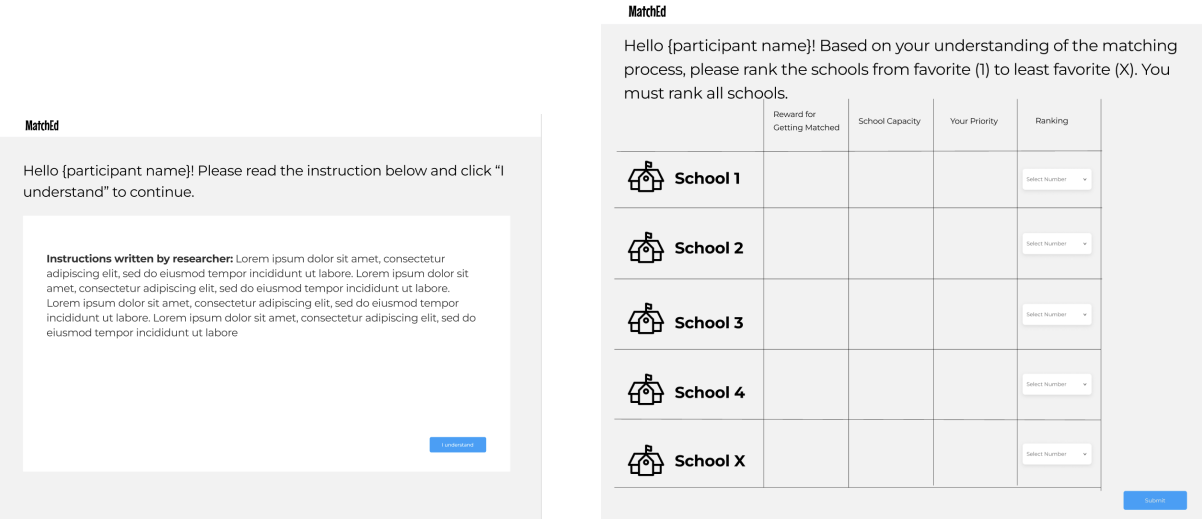
to render React Server Components[5] and reverse proxy requests to the FastAPI backend, which runs on a different origin. The reverse proxy ensures authentication tokens cannot be exploited by malicious actors on the client side.[6]

I configured continuous deployment (CD) pipelines for both the frontend and the backend through GitHub integrations with Vercel and Render. Vercel, the company behind NextJS, provides free hosting with first-class support for NextJS' newest features. Render, a platform-as-a-service akin to Heroku, hosts the FastAPI backend.[7]

Application Design

The application has two types of users: *researchers* and *participants*. Researchers can create accounts and login to configure and conduct experiments. Participants do not login and instead complete the online experiment through a publicly exposed web page.

I created mocks of some of the critical web pages in Figma before starting to code. Figure 2 presents the original designs for the public-facing participant view, while figure 3 shows the eventual implementation. Similarly, figure 4 gives a side-by-side comparison of the new condition form design and implementation. The form dynamically expands as researchers add more schools or participants. The frontend interface was built using the CSS utility framework Tailwind CSS.



(a) Instruction Screen

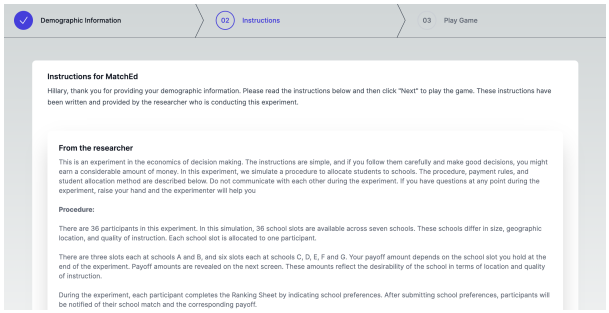
(b) Experiment Screen

Figure 2: Figma Mocks of Participant Views

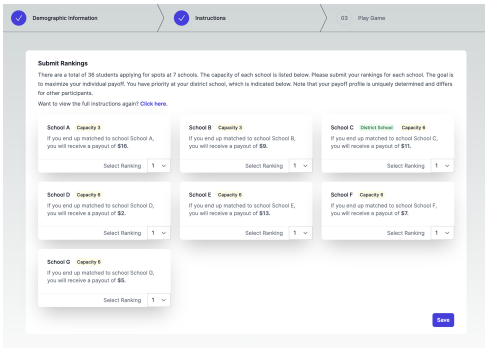
[5]React Server Components (RSC) are a new feature that allow you to render and cache UI components on the server. Read more here.

[6]A reserve proxy is a server—usually at the network edge—that intercepts client requests and sends requests/receives responses from the origin server. The reserve proxy is implemented as a dynamic NextJS API route deployed at the edge via Vercel.

[7]The code for MatchEd is publicly available here.

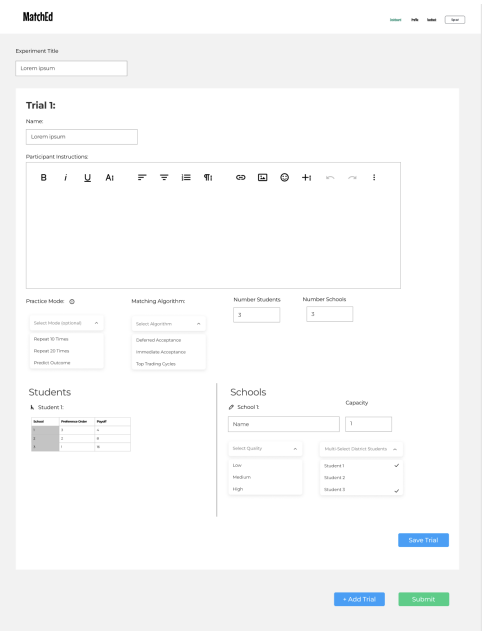


(a) Instruction Screen

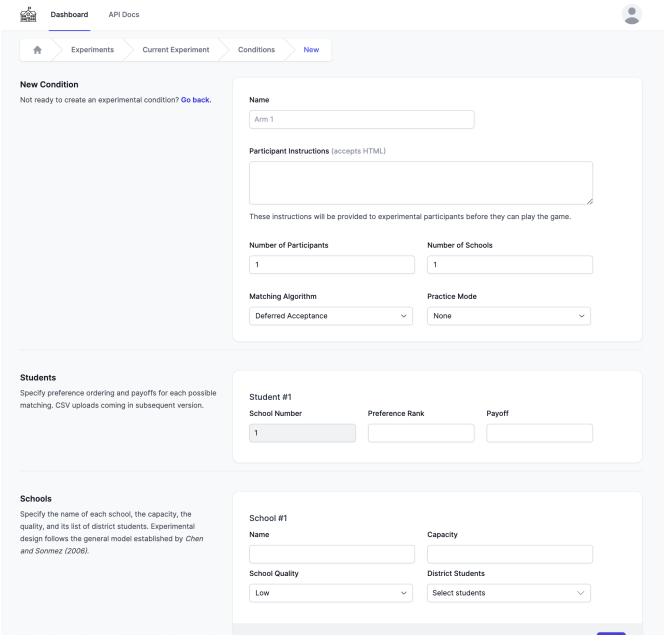


(b) Experiment Screen

Figure 3: Actual Participant Views



(a) Mock of Experiment Form



(b) Actual New Experimental Condition Form

Figure 4: Forms

FastAPI provides auto-generated API documentation based on the OpenAPI specification. The documentation is accessible at the path `/api/docs`, and the full OpenAPI JSON schema is available at `/api/openapi.json`. The docs at `/api/docs` are interactive, allowing users to test the API (see Figure 5). Additionally, users can import the OpenAPI schema into external tools such as Postman to facilitate easy programmatic interaction with the MatchEd platform. The OpenAPI JSON schema can also be used to generate language-specific drivers. The well-documented API interface will prove especially useful for researchers looking to write scripts to query and analyze their data.

2.2 Experimental Design for MatchEd Pilot

To validate the new platform, I conducted a pilot study examining participant learning dynamics in the school choice environment. The experiment consists of two conditions: the control and

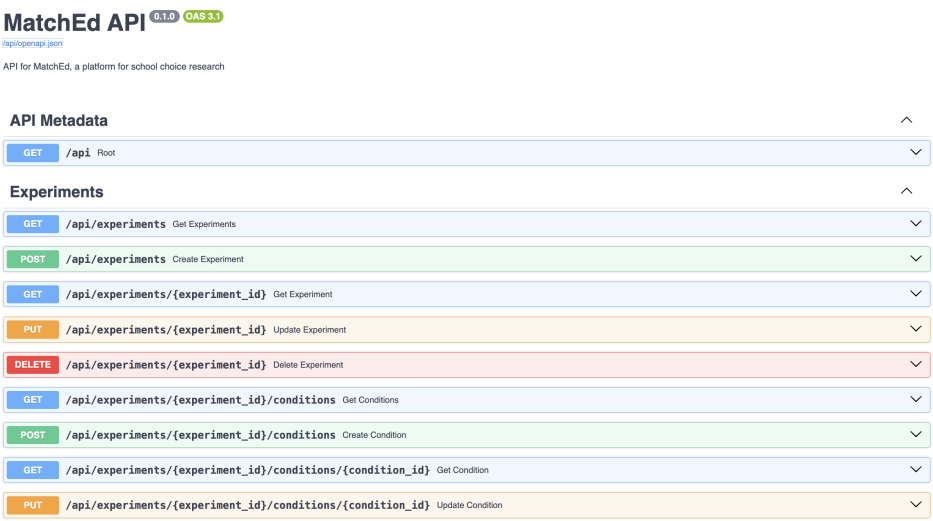


Figure 5: Authenticated users can access interactive API docs

the experimental condition called `repeat`. The control condition is modeled after Chen and Sönmez (2006), who laid the groundwork for subsequent experimental studies on school choice. I leverage their "designed" condition in which 36 participants receive unique payoffs determined by utility functions that account for student proximity to school, school quality, etc. The 36 participants can be matched to one of seven schools: School A - School G. School A and B are "high" quality but can only accommodate 3 students, while schools C - G are "low" quality that can accommodate 6 students. "High quality" schools contribute positively to students' payoff functions, meaning a high proportion of students value School A and B despite there only being 3 spots. This designed environment creates opportunities to analyze different biases in preference submission. [8]

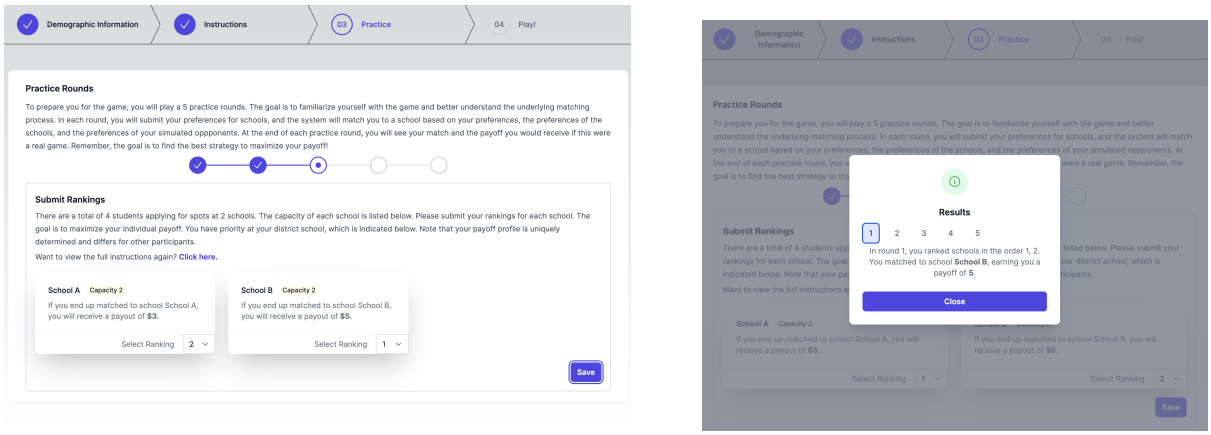
In the control condition, 36 participants were asked to complete the 3-stage online survey via MatchEd. Each participant received a link along with a short disclaimer that the study involved no actual money given it was a pilot. Participants then inputted their demographic information, read the instructions, and submitted their ordered ranking of 7 schools.[9] Given the lack of financial incentive, participants were primarily friends, family, and classmates who volunteered their time to contribute to the research. Most participants completed the online experiment on their mobile devices, and all were invited to contact me, the investigator, with questions. All were instructed to behave as if the stated monetary payoffs were real.

In the experimental `repeat` condition, 36 participants complete the same process as the control condition other than the added "practice stage." The payoff profiles of the 36 participants are the same as the control. After providing demographic information and reading the instructions, participants have 5 chances to submit preferences and observe an outcome. At the end of each practice round, the participant observes the outcome of the matching process (in addition to the outcomes from previous rounds if applicable). The outcomes are determined by running the deferred acceptance algorithm on the participant's submission and the truthful payoff-induced preferences of the other 35 simulated participants. Once they've completed all five practice

[8] The table of payoffs replicated from Chen and Sönmez can be found in the appendix.

[9] Experimental instructions courtesy of Chen and Sönmez (2006) are available through the *Journal of Economic Theory* here.

rounds, participants play the actual game by submitting preferences. A "Thank you" screen indicates they've reached the end of the experiment.



(a) Game 3 of 5 in Practice Mode

(b) Results Modal after Practice Round

Figure 6: Practice Rounds in `repeat` Condition

3 Results

3.1 General Availability of MatchEd

The code for the MatchEd platform is available for open-source use on Github. The code is distributed as-is, and researchers are free to modify the source code for their own particular use cases. For instructions on how to develop MatchEd locally, see the corresponding section in the Appendix.

From the beginning, the goal was to build an extensible platform that could serve a variety of experimental setups in empirical matching mechanism research. As such, researchers are also free to leverage the deployed application at match-ed.vercel.app. The app always reflects the latest code on the `main` branch as continuous deployment (CD) pipelines trigger re-deployments on each push or merge into `main`.

```
$ curl -X GET https://match-ed.vercel.app/api/experiments \
  -H 'Authorization: Bearer <token>'

// Result:
[
    {
        "_id": "6604daa226f7d117f3b7a7c6",
        "name": "Designed Environment",
        ...
    }
]
```

Listing 1: Example cURL Command to MatchEd API

The documentation for the MatchEd API, built with FastAPI and deployed via Render, is accessible to authenticated users here. The API expects either an `access_token` within the request cookies or a JWT token within the `Authorization` headers as a Bearer token. The

`access_token` is stored as an `HTTPOnly` cookie on the frontend to limit security risks. For users interested in programmatically querying the API from a script, copy the `access_token` from your browser console and paste it into the request headers.

3.2 Pilot Study Data

Control Condition

In the `control` arm, I recruited 36 participants from friends, family, and peers. Of the 36 recruited participants, 28 completed all three stages (demographics, instructions, and preference submission). The researcher can track the status of responses in real-time from the MatchEd dashboard using the badges next to the participant number.

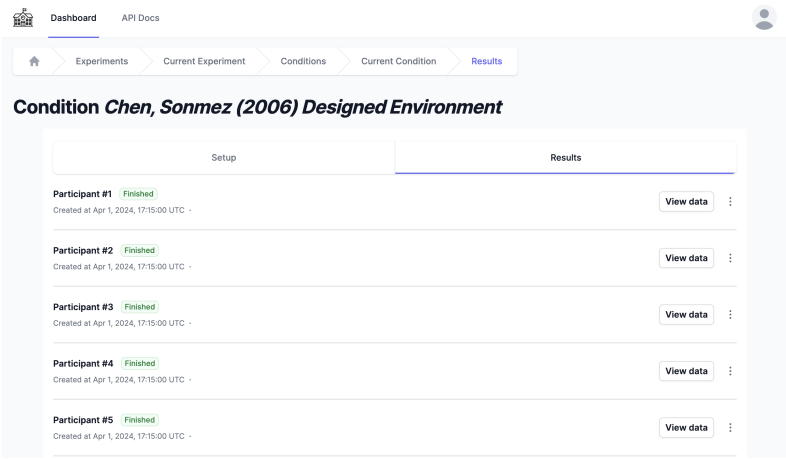


Figure 7: Results dashboard showing completed games

Following the analysis of Chen and Sönmez, I classify each response as either a truthful or an untruthful preference revelation. Truthful preference revelations are those where the submitted ordering is equal to the original ranking up to and including the district school. I exclude any school after the district school since they are irrelevant under DA in the "designed" setting. Untruthful preference revelations are classified into three categories (which are *not* mutually exclusive):

- district school bias (DSB) → A participant puts their district school into a higher position than that in the true preference order.
- small school bias (SSB) → A participant puts the two low-capacity schools —A or B (or both)—into lower positions than those in the true preference ordering.
- similar preferences bias (SPB) → A participant puts schools with the highest payoffs into lower positions.

Participants who submit untruthful preferences are likely to exhibit multiple types of bias. For example, since schools A and B are highly ranked for all participants because they are classified as "good" schools in the experimental design, prioritizing the district school (DSB) often implies a de-prioritization of school A or B (SSB). Table 3 reports the proportion of respondents in one of 9 mutually exclusive and exhaustive categories. The DSB, SSB, and SPB categories in the table imply the respondent *only* exhibited that particular bias and no other.

Approximately 36% of participants reported preferences truthfully, which is lower than the 72% recorded in Chen and Sönmez (2006). I hypothesize that the decrease in the proportion of truthful respondents arose for two primary reasons: First, the pilot study involved no actual money, so the strategic incentives of all participants cannot be guaranteed. Second, the online format reduces the amount of time participants spend internalizing the instructions and the description of the algorithm. Many participants asked "Do I need to read all of this?" when they navigated to the instructions screen. The experimental instructions, which are drawn from Chen and Sönmez, are listed in the Appendix. Multiple participants ($n = 3$) inputted their demographic information but declined to proceed once greeted with long list of instructions, highlighting the problem of response abandonment in crowd-sourced, survey-based research.

The Chen and Sönmez in-person experiment allowed participants 20-25 minutes to read instructions and ask questions before submitting their rankings. Since families submit their schooling preferences via an online form similar to MatchEd, these results raise important questions about how school districts' online platforms might encourage or discourage truth-telling behavior through their instructions. As stated in the literature review, Guillen and Hakimov (2017) found that simple, top-down advice on truth-telling as a weakly dominant strategy contributed most positively to optimal play—participants who read longer descriptions of the algorithm's mechanics played their dominant strategy less. Though drawn from a very small sample, my results seem to support Guillen and Hakimov, and they suggest that instructions are central to determining participants' behavior.

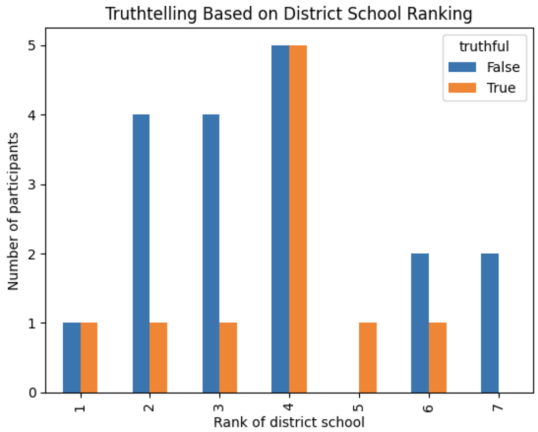


Figure 8: Count of truthful participants per rank of district school

In Chen and Sönmez, the most common form of untruthful preference revelation was SSB & SPB, which accounted for 16.7% of respondents. I observed a similarly high coincidence of SSB & SPB (21.4%). Neither results are surprising given that, by construction, SSB often implies SPB. As stated above, school A and school B are deemed "high" quality schools, making the average payoff across all participants for schools A and B higher than other schools: 17 students value school A the most, while only 1 student values school G the most. As such, moving A or B down in the rankings often coincides with moving a high-ranking school into a lower position.

Figure 8 shows the number of truthful and untruthful students for each possible district school rank. That is, students whose district school aligns with their i-th most preferred school end up in bucket i. We should expect that students with a higher ranking district school would be more likely to tell the truth because they are less susceptible to district school bias[10]. In

[10]Another explanation for this result is our definition of "truthful": Being truthful only requires reporting

particular, we expect having the district school as the top choice to perfectly predict truthtelling behavior since the participant should recognize that they are guaranteed the highest payout. One participant did not exhibit this behavior, reflecting either a very poor understanding of the underlying mechanism or a lack of strategic thinking. The results illustrate that, even when the weakly (strongly in this particular case) dominant strategy is quite obvious, participants can still behave sub-optimally.

Table 3: Rates of Truthful/Untruthful Preferences

	Proportion Truthful		
	(Control)	(Repeat)	p-value
Truthful	0.357	0.57143	0.133
DSB	0.0714	0.04762	0.732
SSB	0.0714	0.00	0.21
SPB	0.00	0.00	NaN
DSB & SSB	0.00	0.00	NaN
DSB & SPB	0.00	0.00	NaN
SSB & SPB	0.214	0.143	0.527
DSB, SSB & SPB	0.25	0.191	0.626
Other	0.0357	0.04762	0.835

Note: We observe the largest difference in proportions between conditions in the truthful category. The t-test for equality of proportions yields a p-value of 0.133, which is not significant. However, the power of the test is severely limited by the small sample size.

Repeat Condition:
As stated above, the `repeat` condition leveraged the same experimental setup as the control, except participants had to work through an additional "practice" step in which they played the game 5 times against 35 simulated participants. All simulated participants played their dominant strategy to reduce the amount of noise participants received from the feedback. During the practice round, participants faced the same payoff profile as the actual game.

21 classmates were recruited to partake in the experiment. Recruitment proved more difficult in the `repeat` condition because of the added time required. Most participants completed the control condition in about 5 minutes, while the `repeat` condition took participants an average of 10-15 minutes.

The practice rounds reveal immediate improvement in participant behavior on multiple fronts. First, in the control condition, some students who had their district school as the highest payoff school still did not submit truthfully. This obvious sub-optimal play disappeared in the `repeat` condition as these particular participants had the opportunity to understand that truthtelling would guarantee the highest payoff of $16.

The overall rate of truthtelling also increased from 35.7% to 57.14%, while the rate of district school bias and small school bias decreased. The SSB & SPB category and the DSB, SSB, & SPB category decreased marginally in the repeat condition as well. Importantly, no bias category increased in the `repeat` condition compared to the control. In a two-sided t-test for

preferences in decreasing payoff order up to and including the district school

equality of proportions between rates of truthtelling, we obtain a test statistic of 1.53, resulting in a p-value of 0.13. Had we observed the same rates of truthtelling with full responses from all 36 participants, the p-value would fall to 0.06. Given the struggle to recruit participants to a free study, we expect a larger-scale paid experiment would result in a statically significant difference at the 5% level.

A key motivation for the experiment is to better understand learning dynamics among participants in the `repeat` condition. In particular, we are concerned with how quickly participants converge to the dominant strategy. To analyze this behavior, we look at the proportion of participants who play truthfully across the 5 practice rounds and the actual game. As in Ding and Schotter (2017), we find that the fraction of subjects reporting truthfully increases over time. Although our result does not have strict monotonicity, the increase from an initial truthtelling proportion of 0.29 in practice round 1 to an final truthtelling rate of 0.57 is statistically significant with a p-value of 0.06.

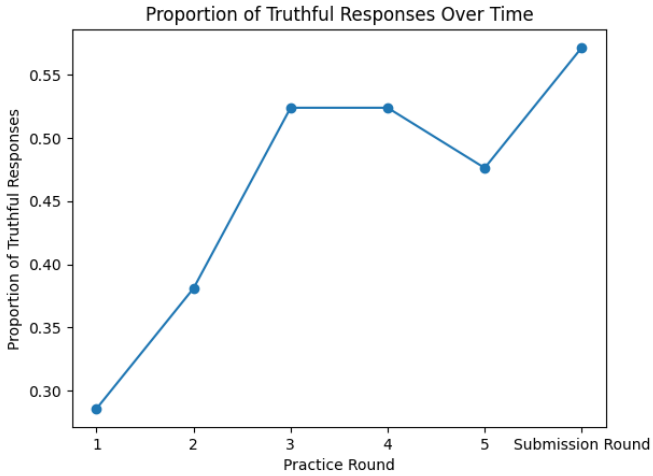


Figure 9: Proportion of truthful responses over time

The slight decrease in truthtelling in the final practice round may be explained by the difference in experimental setup between Ding/Schotter and the current experiment. Ding and Schotter told participants that they would receive the payoff from one randomly selected round, while our experiment modeled the practice rounds as purely for exploration—the payout is determined by the final submission, meaning it is independent of behavior during the practice rounds. This introduces new strategic considerations as participants may want to use the practice rounds to explore their strategy space, even if they suspect a strategy may not lead to a higher payoff. This setting better models the real-world environment in which families may look to interact with the mechanism before playing the one-shot game issued by the school district.

Table 4: Average Payoff

Control	Repeat	t-stat (p-value)
10.61	12.33	2.22 (0.03)

Note: The difference in average payoff between the two experimental conditions is statistically significant, providing evidence of convergence to dominant strategy equilibrium (DSE).

The final component of our analysis is to compare the efficiency of outcomes between the `control` and `repeat` conditions. Since we were unable to recruit 36 participants for both experimental conditions, we assumed truthful preference revelation in all unplayed games. We took this approach since all experimental participants played (i.e. internalized school capacity information) as if 36 students were vying for 36 spots. We evaluate the efficiency of the mechanism by examining the mean payoff. In the `repeat` condition, the average payoff among experimental participants was $12.33, while in the `control` condition, the average payoff was $10.61. The difference of means ($1.72) is statistically significant at the 5% level. Our results align with previous work on the efficiency of the Gale-Shapley deferred acceptance mechanism. That is, the outcomes of repeated games played 10 or less times tend to be more efficient than the one-shot alternative.[11]

4 Future Work

Extensions to MatchEd Platform

Although MatchEd was designed to support a variety of matching experiments, it could be extended to improve the user experience. First, the application could offer more analytics about participant behavior. Information on elapsed time during repeated games or even user clicks could prove beneficial for analysis. Other opportunities to improve the platform include built-in data visualizations akin to those in figures 8 and 9. Second, the application currently requires significant manual data entry when creating a new experiment. Although users can create resources via the REST API, we hope to provide a CSV upload feature to streamline data entry on the client-side.

The biggest shortcoming with the current platform is that it is not conducive to crowd-sourced research since every participant plays the game through a personalized link. This makes it quite difficult to recruit participants since researchers cannot distribute a single survey link on social media or other platforms. Directing users to unplayed games through a single link poses significant technical challenges, especially when considering the possibility for race conditions. Ideally, we can design all API requests to be idempotent, meaning that making the same request multiple times will have the same effect as making it once.

Additional Experiments and Analysis

With more time and money, we would have liked to compare the learning dynamics in five rounds against ten or more rounds (as is done in pre-existing literature) to assess whether there is a statistically significant difference. Additionally, we want to build on the current study's results by examining behavior when the game (i.e. payoff profile) changes in each practice round. When the payoff profiles change, participants must develop a fundamental understanding of the mechanism to achieve higher payoffs over time. In the current experiment, participants could use the practice rounds for a "guess-and-check" approach in which they tried different preference orderings without considering the mechanism at all. We could also ask participants in the repeated condition to write their understanding of the optimal strategy upon completing the experiment. We could analyze these responses with natural language processing tools to answer the question of whether people fundamentally understand the weakly dominant strategy.

[11] Ding and Schotter compare the repeated game to an intergenerational advice setting, but the same result still holds.

With more data, we also could have conducted additional analysis. Specifically, we were interested in leveraging logit or probit models to investigate the impact of various factors (such as ranking of district school) on the probability of truthtelling. Unfortunately, with such a limited sample, these models produce unstable estimates and large standard errors. With a predictive model, we could also generate simulated data to further expand the dataset.

5 Conclusion

This project contributes to the experimental school choice literature in two ways: First, the new online platform MatchEd offers a flexible interface through which researchers can design and conduct experiments in the school choice setting. Second, a pilot study conducted with MatchEd illustrates the potential benefits of playing simulated matching games to reduce the rate of suboptimal preference manipulation among participants.

Despite the use of strategyproof algorithms, preference manipulation remains common in the school choice setting. Reducing preference manipulation has important benefits for the long-term stability of the school choice market, and it saves families time otherwise spent debating their strategy. Although research on the theoretical guarantees of matching algorithms is well-established, we hope this paper sparks further policy-driven research on how to increase rates of truthtelling. As the field of school choice matching shifts towards behavioral economics, MatchEd can offer a model for how to conduct large-scale, online experiments that seek to better understand market participants' decision-making processes.

6 References

[1] Abdulkadiroglu, Atila & Tayfun Sönmez (2003). "School Choice: A Mechanism Design Approach." *American Economic Review*, 93(3): 729–47.

[2] Calsamiglia, Caterina, Guillaume Haeringer & Flip Klijn (2010). "Constrained School Choice: An Experimental Study." *American Economic Review*, 100 (4): 1860-74.

[3] Celebi, O. (2022). "Best-response dynamics in the Boston mechanism". *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.4254039

[4] Chen, Yan & Tayfun Sönmez (2006). "School Choice: An Experimental Study." *Journal of Economic theory*, 127(1): 202–31.

[5] Ding, Tingting & Schotter, Andrew (2019). "Learning and Mechanism Design: An Experimental Test of School Matching Mechanisms with Intergenerational Advice," *The Economic Journal, Royal Economic Society*, vol. 129(623), pages 2779-2804.

[6] Guillen, Pablo & Hing, Alexander (2014), "Lying through their teeth: Third party advice and truth telling in a strategy proof mechanism", *European Economic Review*, 70, issue C, p. 178-185.

[7] Guillen, Pablo & Hakimov, Rustamdjan (2017), "Not quite the best response: truth-telling, strategy-proof matching, and the manipulation of others", *Experimental Economics*, 20, issue 3, p. 670-686.

[8] Guillen, Pablo & Hakimov, Rustamdjan (2017), "The effectiveness of top-down advice in strategy-proof mechanisms: A field experiment", *European Economic Review*.

[9] Leshno, J., & Lo, I (2017). "The simple structure of top trading cycles in school choice". *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.2954870

[10] Pathak, Parag A. & Tayfun Sönmez. 2013. "School Admissions Reform in Chicago and England: Comparing Mechanisms by Their Vulnerability to Manipulation." *American Economic Review*, 103 (1): 80-106.

[11] Roth, A.E. (2008). "Deferred acceptance algorithms: history, theory, practice, and open questions," *Int J Game Theory* 36, 537–569.

[12] Roth, Alvin E. (1982). "Incentive compatibility in a market with indivisible goods". *Economics Letters*. 9 (2): 127–132.

[13] Shapley, Lloyd & Scarf, Herbert (1974). "On cores and indivisibility". *Journal of Mathematical Economics*. 1: 23–37.

7 Appendix

Supplementary Information on Matching Algorithms

Stability of DA: The proof proceeds by contradiction. Assume that matching M creates an unstable matching between students S and schools U. Let $(s, u) \in S \times U$ be the blocking pair in M. This implies that $u \succ_s M(s)$ (student s prefers u to its current assignment). Student s must have proposed to u before proposing to $M(s)$ since students propose in preference order. The blocking pair also implies that $s \succ_u M(u)$. These two implications establish a contradiction: s must have proposed to u and been rejected despite u preferring s to its already-matched students.

Top Trading Cycles Algorithm: Attributed to David Gale in a paper by Shapley and Scarf in 1974 that addresses the "house allocation problem," the top trading cycles algorithm (TTCA) is strategyproof and, unlike DA, produces Pareto efficient matchings. However, TTCA does not guarantee stability. The New Orleans school district, among others, currently leverages TTCA in its school match. Let $\mathcal{S}$ be the set of all students, and let $\mathcal{U}$ be the set of all schools. Similarly, let S and U be the sets of unassigned students and schools respectively.

Algorithm 3 Top Trading Cycles Algorithm

Initialization:
$S \leftarrow \mathcal{S}$
$U \leftarrow \mathcal{U}$
repeat
  Each $u \in U$ draws a directed edge to its favorite remaining student.
  Each $s \in S$ draws a directed edge pointing to its favorite remaining school.
  Select a directed cycle and match each s to the school u to which they point.
  Update U and school capacities.
until $U = \emptyset$ or all schools are filled

Instructions for Running Code Locally

The code can be downloaded here. The application is structured as a "monorepo" with the NextJS application stored within the `frontend` directory and the FastAPI application stored within the `backend` directory.

Running the Backend Server: First, make sure you are running Python 3.10 or above. Then, `cd` in to the `backend` directory and install the dependencies with `pip3 install -r requirements.txt`. Next, you'll need to create a `.env` file to store environment variables and secrets. Do not commit this file to Git to protect sensitive data. The following environment variables are required:

- `ALGORITHM` → the algorithm used to generate JWT tokens, usually "HS256"
- `FRONTEND_URL` → the URL for the NextJS application, likely "http://localhost:3000" for local development
- `JWT_DURATION` → the duration of the JWT token in minutes until the user is forced to login again
- `MONGO_URI` → connection string for your MongoDB database, either running locally or in the cloud via MongoDB Atlas
- `MONGO_DB_NAME` → the databse within your MongoDB cluster that contains application data
- `SECRET_KEY` → the secret key used to encrypt JWT tokens

The easiest way to spin up a MongoDB cluster is via MongoDB Atlas, their cloud database-as-a-service offering. You should now be able to run the backend server with `python3 -m uvicorn main:app --reload`. The FastAPI server should be running on port 8000 by default.

Running the NextJS Server: `cd` into the `frontend` directory and install dependencies with `npm install`. Next, you'll need to create a `.env` file to store environment variables:

- `NEXT_PUBLIC_BACKEND_URL` → the FastAPI server URL, probably "http://localhost:8000" for local development
- `NEXT_SERVERLESS_API` → the NextJS serverless API url, probably "http://localhost:3000" for local development

Start the server with the command `npm run dev`.

Participant Instructions

`Control` Condition:

This is an experiment in the economics of decision making. In this experiment, we simulate a procedure to allocate students to schools. The procedure, payment rules, and student allocation method are described below. Do not communicate with each other during the experiment. If you have questions at any point during the experiment, contact the experimenter for help.

Procedure:

There are 36 participants in this experiment. In this simulation, 36 school slots are available across seven schools. These schools differ in size, geographic location, and quality of instruction. Each school slot is allocated to one participant.

There are three slots each at schools A and B, and six slots each at schools C, D, E, F and G. Your payoff amount depends on the school slot you hold at the end of the experiment. Payoff

amounts are revealed on the next screen. These amounts reflect the desirability of the school in terms of location and quality of instruction.

During the experiment, each participant completes the Ranking Sheet by indicating school preferences. After submitting school preferences, participants will be notified of their school match and the corresponding payoff.

Allocation Method:

In this experiment, participants are defined as belonging to the following school districts. A priority order is determined for each school. High priority students are those who belong to the school's district. Low priority students are those who do not belong to the school's district. The priority among the Low priority students is determined randomly to be fair.

The allocation of school slots is obtained as follows:

1. An application to the first ranked school in the Ranking Sheet is sent for each participant.
2. Throughout the allocation process, a school can hold no more applications than its number of slots. If a school receives more applications than its capacity, then it rejects the students with lowest priority orders. The remaining applications are retained.
3. Whenever an applicant is rejected at a school, his application is sent to the next highest school on his Ranking Sheet.
4. Whenever a school receives new applications, these applications are considered together with the retained applications for that school. Among the retained and new applications, the lowest priority ones in excess of the number of the slots are rejected, while remaining applications are retained.
5. The allocation is finalized when no more applications can be rejected. Each participant is assigned a slot at the school that holds his/her application at the end of the process.

An example of this process is available here. These instructions will be available for re-reading on the next screen as you input your rankings.

Note: The instructions in the *repeat* condition are identical except one sentence is added: "Before submitting school preferences, you will have 5 opportunities to practice playing the game under the same conditions."

Payoff Profiles

The following table, replicated from Chen and Sönmez (2006), shows the payoff profiles for each of the 36 participants. Bold numbers indicate that the student lives in the district of the corresponding school.

Table 1
Payoff table in the designed environment

	Schools						
Student ID	A	B	C	D	E	F	G
1	**13**	16	9	2	5	11	7
2	**16**	13	11	7	2	5	9
3	**11**	13	7	16	2	9	5
4	16	**13**	11	5	2	7	9
5	11	**16**	2	5	13	7	9
6	16	**13**	7	9	11	2	5
7	13	16	**9**	5	11	7	2
8	16	9	**11**	2	13	7	5
9	16	13	**2**	5	9	7	11
10	16	7	**9**	5	2	11	13
11	7	16	**11**	9	5	2	13
12	13	16	**9**	11	2	7	5
13	9	16	2	**13**	11	5	7
14	16	5	2	**9**	7	13	11
15	13	16	9	**11**	2	7	5
16	16	13	11	**5**	9	7	2
17	13	16	5	**7**	2	9	11
18	16	13	5	**9**	7	11	2
19	11	16	7	5	**13**	9	2
20	16	13	7	9	**5**	2	11
21	13	16	2	7	**9**	11	5
22	16	11	7	2	**9**	5	13
23	16	13	7	2	**5**	11	9
24	16	13	11	5	**9**	2	7
25	13	16	2	5	11	**9**	7
26	16	13	5	9	7	**2**	11
27	7	11	5	2	13	**9**	16
28	16	13	7	2	11	**5**	9
29	7	11	16	13	2	**9**	5
30	16	9	7	2	5	**11**	13
31	11	16	7	2	5	9	**13**
32	13	9	16	2	5	7	**11**
33	13	16	11	9	7	5	**2**
34	16	11	2	7	5	13	**9**
35	7	16	2	5	11	13	**9**
36	16	13	5	7	9	2	**11**

Figure 10: Payoffs

Data Legend

The CSV file `control_processed.csv` contains data from the control condition. The data was extracted from MongoDB and transformed into a format more conducive to analysis:

Column	Explanation
student_id	ID of the participant
complete_truthtell	1 if t1-t7 = p1-p7, 0 otherwise
partial_truthtell	1 if truthful and submitted preferences true up to district school
dsb	indicator for district school bias
ssb	indicator for small school bias
spb	indicator for similar preference bias
DsbSSb	indicator for district school bias and small school bias
DsbSPb	indicator for district school bias and similar preference bias
SsbSPb	indicator for small school bias and similar preference bias
DsbSSbSPb	indicator for district school bias, small school bias and similar preference bias
DsbOnly	indicator for district school bias only, meaning no small school bias or similar preference bias
SsbOnly	indicator for small school bias only, meaning no district school bias or similar preference bias
SpbOnly	indicator for similar preference bias only, meaning no district school bias or small school bias
r1	ID of the school ranked first by the participant
r2	...second...
r3	...third...
r4	...fourth...
r5	...fifth...
r6	...sixth...
r7	...seventh...
t1	ID of the school ranked first in true preference ranking
t2	...second...
t3	...third...
t4	...fourth...
t5	...fifth...
t6	...sixth...
t7	...seventh...
p0	Payoff for school ID 0
p1	...1
p2	...2
p3	...3
p4	...4
p5	...5
p6	...6
district_school	ID of the district school

Similarly, the CSV file `experimental_processed.csv` contains data from the experimental condition of the pilot experiment. All data analysis is contained within the `analysis` directory in the project repository. `control_arm.ipynb` has the analysis for the control condition, while `repeat_arm.ipynb` has the analysis for the `repeat` condition.