Making Harvard Dining More Convenient Through an Adapted TTC Mechanism

Caroline Abel*, Ben Altschuler*, Charles Onesti*

December 6, 2021

* All students contributed equally to all parts of the project.

1 Introduction

Ask any upperclassman and they will likely tell you that they miss the atmosphere of their First-year meals at Annenberg where they made friends and shared ideas with hundreds of fellow classmates. Upperclassman housing decisions can create arbitrary divides between students and an 8 person blocking group offers few guarantees. We imagine a system where we can balance feasibility constraints of upperclassman house dining with greater freedom for students to mix between house dining halls.

We decided to tackle the problem of restrictive Harvard dining hall access through a modified Top-Trading Cycle mechanism (TTC). In this paper we will start by talking about the TTC Mechanism as described in class and then introduce a modified TTC mechanism that is able to maintain strategy-proofness and Pareto-optimal matchings across non-strict preference orderings. Finally, we will apply this modified TTC algorithm to assign dining halls to students that are at least as preferred as eating in their own dining hall. Throughout the paper we will relate our mechanism design choices to the specific regulations and dynamics of the Harvard house dining system. We will test our mechanism through a Python implementation using four different simulated populations which we think are representative of an average weekday lunch, weekday dinner, weekend lunch, and weekend dinner. We then will provide results, analysis, and reflections on the mechanism's potential future impact.

1.1 Motivation

We are motivated by our experience trying to eat in dining halls other than our own during lunch and dinner. Naturally, students spend their days outside of their house and often do not have time to travel back to their respective house between classes or meetings. We also want to bring back that Annenberg experience for upperclassmen living in houses.

On the part of Harvard University Dining Services (HUDS), we are motivated to give greater predictability and stability to the volume of students eating at each dining hall with meal specific data. We also collect people's true preferences before allocation of dining hall swipes and can learn who prefers to eat where but was unable to go. Access to more accurate volume estimates for each dining hall will improve the dining and work experiences for both parties. The information could also improve consumption estimates leading to less food waste in the dining halls.

1.2 Problem Statement

The challenge we take on is to design an automated mechanism which given reports of housing preferences from students, finds a weakly more beneficial matching of dining hall access to students. This is a one-sided matching problem where students have strict preferences over which houses to eat in and dining halls do not have preferences over students but will want to have constraints on capacity. This problem is similar to the school choice

problem discussed in Marek Pycia and M. Utku Unver's "Trading Cycles for School Choice," which we drew inspiration from[1]. In the problem, there is a set of schools H and each school h has q_h copies representing seats in the school. Pycia and Unver use TTC to find a Pareto efficient and strategy proof solution to students' strict preference over the set of schools H and weak preference over seats in a school.

Ideally, our mechanism would have several of the desirable mechanism properties we learned about in class. For our purposes, we are also interested in strategy-proofness for students, Pareto-efficiency, individual rationality and a core-stable outcome. The TTC mechanism is a good fit for securing these properties as proved by Alvin Roth et al and which in our textbook states that with strict preferences, TTC is strategy proof, Pareto-efficient, and core.[2,3] Our problem complicates the general TTC mechanism as students are indifferent about which other student, who currently own their most preferred house, to point to in the TTC. Put another way, students have preferences across houses but not across students in a given house.

2 Our Solution

2.1 Mechanism Description

For our modified TTC we will provide each student in the system with an initial house. This corresponds to the house that they are in. Students in a given house are always allowed to eat in their house for all meals (unless they trade for a more preferred house in the TTC mechanism). Students must provide a strict preference ordering over dining halls for a given meal which can include their own house. If they do not include their own house it will be added as their final preference. We think students will have some sort of preference over houses because of proximity or friends in certain houses, so we can force this strict preference. The challenge presented is that students are indifferent between trading with any agent from the same house. Our TTC instance has agents pointing to another agent who has their most preferred house. It is not fully defined how we should handle the case of choosing an agent within a house to trade with. It is not strategy proof for each individual to simply pick at random an agent in their most preferred house to trade with. This indifference problem is addressed by Daniela Saban and Jay Sethuraman in a paper they wrote about indifference in TTC for housing allocation titled "House Allocation with Indifferences." They presented the problem as follows:

> To illustrate, consider two agents, 1 and 2 endowed with objects a and b respectively. Suppose agent 1 is indifferent between objects a and b, whereas agent 2 strictly prefers a to b. To guarantee Pareto efficiency, the agents must trade their endowments, but an arbitrary tie-breaking rule may rank a before b for agent 1, resulting in the inefficient allocation where agents do not trade. [4]

Pareto efficiency is at stake when we have to deal with indifference in preference orderings. Our mechanism requires a way to tie-break and make sure that agents agree to uniformly settle their indifference. The problem of which student to point to in a given house is solved by a random priority ranking across all students that is decided on before the TTC mechanism is complete. As shown by Saban and Sethuraman such a modification to TTC maintains the desirable properties of strategy-proofness and finding matchings that are in the weak core. In practice, this priority ordering will specify that for each agent there

[1]Pycia, Marek and Unver, Utku, Trading Cycles for School Choice (July 31, 2011). Available at SSRN: https://ssrn.com/abstract=1899344 or http://dx.doi.org/10.2139/ssrn.1899344

[2]Roth, Alvin E. (1982-01-01). "Incentive compatibility in a market with indivisible goods". Economics Letters. 9 (2): 127–132. doi:10.1016/0165-1765(82)90003-9.

[3]2021 Parkes & Seuken

[4]Saban, Daniela, and Jay Sethuraman. "House Allocation with Indifferences." Proceedings of the Fourteenth ACM Conference on Electronic Commerce, 2013, https://doi.org/10.1145/2482540.2482574.

is exactly one other agent whom they strictly prefer in that round of TTC. Since agents are strict already when it comes to houses, the priority ordering steps in for within house tie-breaking. Given the above formulation the computation of the mechanism follows the procedure of TTC as described in the textbook. Once a cycle has formed all agents in the cycle are assigned the house they point to and then are removed from our list of agents who are yet to be assigned a house. The mechanism finishes when each agent has been assigned their most preferred feasible matching given their priority rank. One interesting note about our mechanism design is that preference orderings across all students enforce the fact that on each iteration of TTC only the top student among all the students in a house can trade. This means that a maximum of 12 people can trade out of the system in each round. Although it takes slightly longer to resolve than a traditional TTC, the algorithm remains polynomially time bounded in the number of students.

This mechanism will run for every lunch and dinner during the week except for Thursday dinner in which students will not be able to trade because of Community Night, the night each week when no interhouse eating is allowed. Students will swap dining hall swipes only with students who want to swap for that same meal and day. A functioning production version of this mechanism would have students report dining preferences from client applications (mobile or web app) to a central server running the mechanism and where each dining hall's HUDS staff can set the volume, and other meal parameters. The results of the TTC matching could be automatically communicated back to the clients and to the swipe scanner machines using Harvard IDs. Capturing the complexity of the dining restrictions that HUDS imposes is important for our mechanism to be able to effectively replace the current system of rigid constraints. The following sections address these more complex restrictions by using generated Dummy agents who interact in the TTC mechanism like normal agents but who do not represent a real person.

2.2 Sister Houses

Currently in the dining system, each Quad house has a River "Sister house" where quadlings can eat during the week and weekend without restrictions. Students from Pforzheimer can eat in Adams, students from Cabot in Lowell, and students from Currier in Winthrop. To incorporate this feature of the dining hall system in our modified TTC solution, we allowed each River House with a Quad Sister house to release a certain number of Dummy agents which will have their top and only preference as their respective Quad Sister house. For example, Adams House can release 200 Dummy agents whose top and only preference is Pforzheimer House so that students in Pforzheimer can come eat in Adams or trade this more desirable River House swipe to eat in another house. People in other houses might end up eating in Adams instead of Pforzheimer students but regardless the spots were offered to Pforzheimer students.

In order to prevent Dummy agents from trading with each other, they are ranked last relative to real students. Dummy agents who trade with each other before real students is a problem because no real matchings were made and it could prevent a real matching at later rounds in the TTC. Lower priority means that the Dummy agents will trade after all of the students of their house if necessary, so this problem is avoided.

2.3 Guest Swipes

To allow for house residents to invite friends for a meal, the interhouse dining restrictions allow each resident a plus one at certain dates and times. For example, one third of houses have a plus one policy which allows their residents one invite for a guest at lunch time and half of the houses have this policy for dinnertime. This is an open offer from the dining hall and it has no guarantee of how many of their residents will actually use their plus one so their volume can at worst double without warning. Our DHall TTC mechanism naturally adapts to capture the expression of guest swipes.

The implementation of our mechanism assigns a unique identifier to every agent in the system. If a dining hall wanted to allow their residents a guest swipe for a certain

meal, they could generate a new Dummy agent for anyone who wants to use a guest swipe. The resident could specify the identifier for another student of another house which they would like the Dummy agent to trade on a cycle with. This will allow the mechanism to have the friend point specifically at the Dummy bypassing the priority system and allowing the Dummy to point anywhere without preference capturing the idea that since the guest is leaving their original house, there is an open spot that can be filled by someone else and the Dummy can end up anywhere without issue including simply doing a two-swap with the guest in the worst case. This method ensures that guests are accounted for and unique, guests cannot leverage a guest swipe to eat somewhere else, and the space opened at the guests initial house is not wasted. Since these specific preferences are handled well by TTC and indifference is addressed by the priority system. Expressing the guest swipe system does not change any of our mechanism properties of strategy proofness, Pareto-efficiency, and core outcomes. Under our time constraints, we were unable to collect simulation results on testing the guest swipe feature, but as shown our mechanism is capable of handling such functionality.

3 Implementation

3.1 Description

We constructed our mechanism implementation so that there exist 6 different types of agents in the system, using the House neighborhoods (groups of 3 Houses) as groupings. QuadAgent, RiverEastAgent, RiverWestAgent, and RiverCentralAgent are agents whose preference ordering is a random permutation over the 3 Quad houses, 3 River East houses, 3 River West houses, and 3 River Central houses respectively. In other words, these agents will have a random preference ordering over a particular neighborhood. The AllRiverAgent is an agent whose preference ordering is a random shuffle over the 9 River Houses. The ParticularAgent is an agent who wants only one particular house or no change in their dining hall swipe for that meal. For all agents, if their own house is not in their preference, it would be considered as directly after their reported preference list by default. We thought that these agents would be representative of the students who participate in the mechanism. Our mechanism allows for students to indicate their own house as a preference for a given meal (like a ParticularAgent who chooses their own house). However, for our simulations we do not include such 'degenerate' agents who are content with their own dining hall.

The 6 agent types are used in this experiment to represent real students. The last agent type is the Dummy agent which does not represent a student. The idea of the Dummy agent class is to provide the HUDS staff with greater control over the system from an administrative point of view. The dining halls themselves would control the number and preferences of these agents allowing them to express different dining permissions and restrictions. As we learned in class, TTC very tightly enforces the balance of resources. Every resource is used and no resource is created. This system is too binding for its practical application in university dining because the current system allows for the flexibility of dining numbers to fluctuate. Dummy agents allow the expression of this flexibility by dialing up or down the creation of swipes. For example, if a student from House A trades with a Dummy from House B, it is as if House B created a new swipe for the student and House A lost that swipe because the Dummy is not very hungry. Through our priority system, Dummy agents will always be able to create their intended imbalances if the dining situation requires it because they are prioritized below normal agents so they trade last.

We built code that solved the TTC instance in our unique formulation. Specifically, we needed to write a program that repeatably created a graph from the agents still unmatched. For each graph instance we found the cycles by looking for the strongly connected components of the graph. We used Kosaraju's algorithm to find SCC's in linear time.[5] Strongly connected components of the graph represent a matching cycle as they indicate

[5] "Kosaraju's Algorithm." Wikipedia, Wikimedia Foundation, 12 Aug. 2021, https://en.wikipedia.org/wiki/Kosaraju%27s_algorithm.

groups of agents that can all follow a path along each other to get to any other agent in the connected component. Once we assign the house of the agent that each agent points to in the connected component we remove those agents from our list of unmatched agents and create a new graph where still unmatched agents can only point to other unmatched agents. If no currently unmatched agent can find an agent with their most preferred house it means that that preference is no longer viable and we pop that from their preference ordering. The mechanism concludes when every agent is assigned a house in their preference order (or their own house if all preferred houses above their own house are not feasible).

An important aspect of our simulation is our model for how the average population of Harvard College students will choose to select their preference orderings for lunch and dinner. We identified four different meal times that are sufficiently different in preferences to ensure the mechanism was flexible enough to find good outcomes for all situations it might be used for in practice. We did not consider breakfast as most students will not eat breakfast or eat in their own house. Only Annenberg and Quincy offer hot breakfasts and they have no restrictions. The four meal times we focus on are weekday lunch, weekday dinner, weeknight lunch, weeknight dinner. For **weekday lunch** we recognize that a lot more students will prefer to eat in Adams and other River Central houses as they are located closest to the Yard. On the other side of the coin there will be very few students requesting to eat in the Quad. There will also be fewer River East and River West agents as they are further from the yard. Students will also be less picky about where they eat and thus there will be fewer particular agents. For **weekday dinner** we introduce more particular agents as people will attempt to make dinner plans with specific individuals in a house. For **weekend lunch** we introduce more particular agents as people are able to make plans to see somebody for lunch on the weekend when they have more time. We also introduce fewer overall agents as many more students will be happy to eat in their own dining hall. Finally for **weekend dinner** we introduce more particular agents as well as lowering the number of neighborhood indifferent agents. This is because we assume students will not usually desire a random house but will likely want to eat in specific houses for events or scheduled dinners.

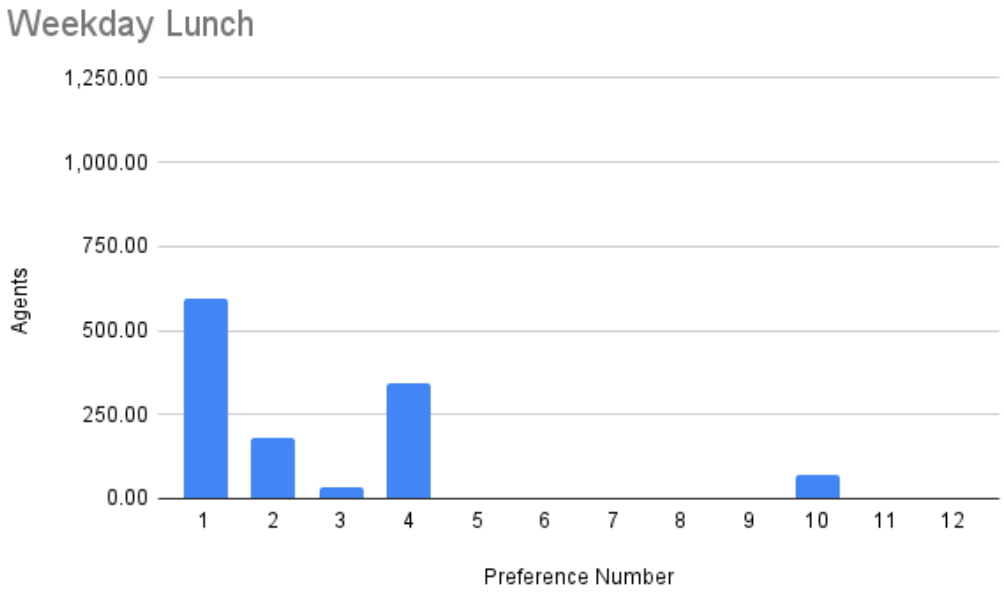
3.2 Results

We ran the simulator on each variant 50 times and averaged the 50 iterations to build each table. Some important metrics that help describe the efficacy of our TTC formulation are 'average improvement', 'number of swaps', and 'average cycle length'. Average improvement is a measure of how much better a student's final assignment is given their preference order and compared to the assignment of their own house. This metric is defined as the index of the students initial house in their preference order subtracted from the index of their assigned house all divided by the length of their preference order minus 1. The bounds of values range from 0 (being allocated your own house) to 1 (getting your first choice, with your own house being your last choice). Number of swaps represents the number of agents that are assigned to a house other then their own. 'Average Cycle length' represents how long each cycle that is longer than 1 is in the running of the TTC mechanism.

3.2.1 Weekday Lunch

Total Number of Agents	Average Total Number of Rounds	Number of Non Dummy agents
1530	268.10	1230

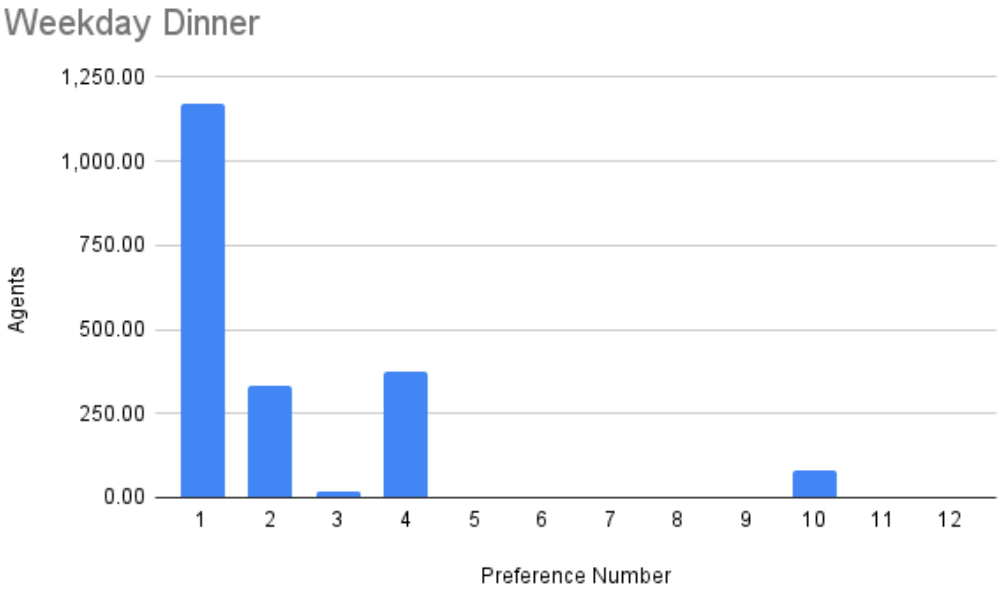
Average Improvement	Average Number of Swaps	Average Cycle Length
0.547	1018.20	3.321



3.2.2 Weekday Dinner

Total Number of Agents	Average Total Number of Rounds	Number of Non Dummy agents
2280	392.70	1980

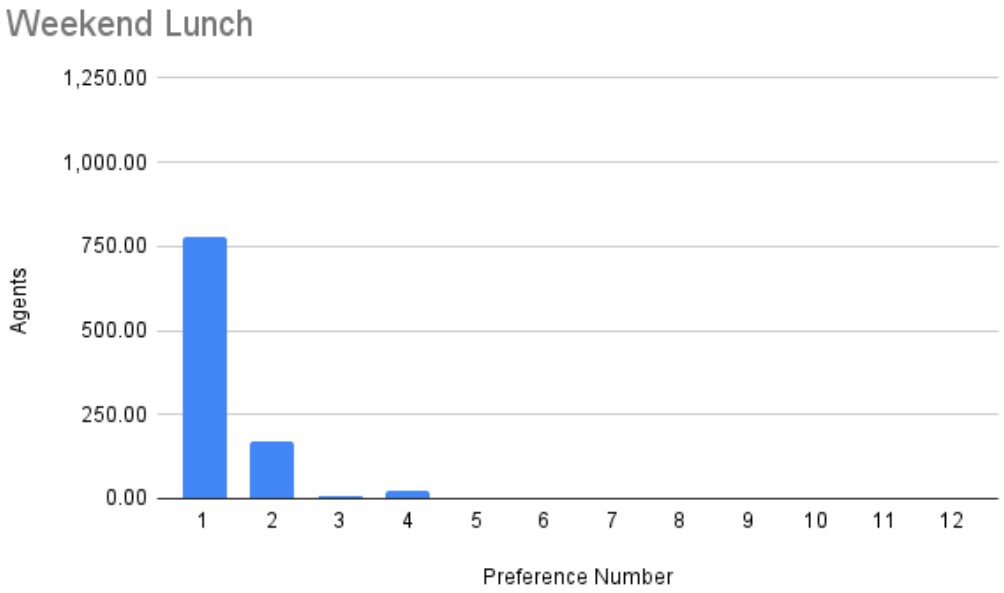
Average Improvement	Average Number of Swaps	Average Cycle Length
0.613	1543.70	3.417



3.2.3 Weekend Lunch

Total Number of Agents	Average Total Number of Rounds	Number of Non Dummy agents
1284	259.80	984.0

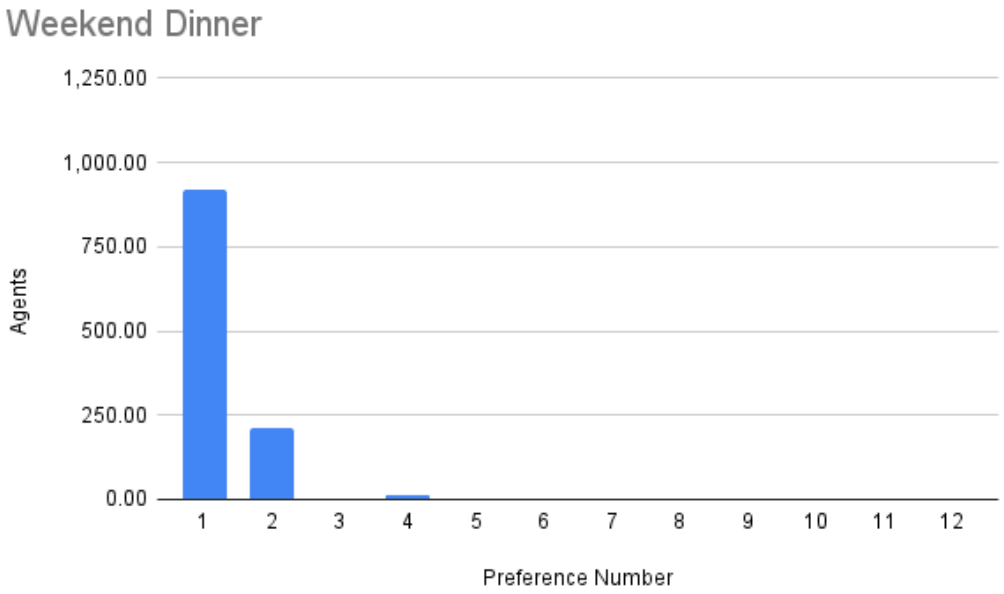
Average Improvement	Average Number of Swaps	Average Cycle Length
0.815	1036.10	3.509



3.2.4 Weekend Dinner

Total Number of Agents	Average Total Number of Rounds	Number of Non Dummy agents
1443	281.4	1143

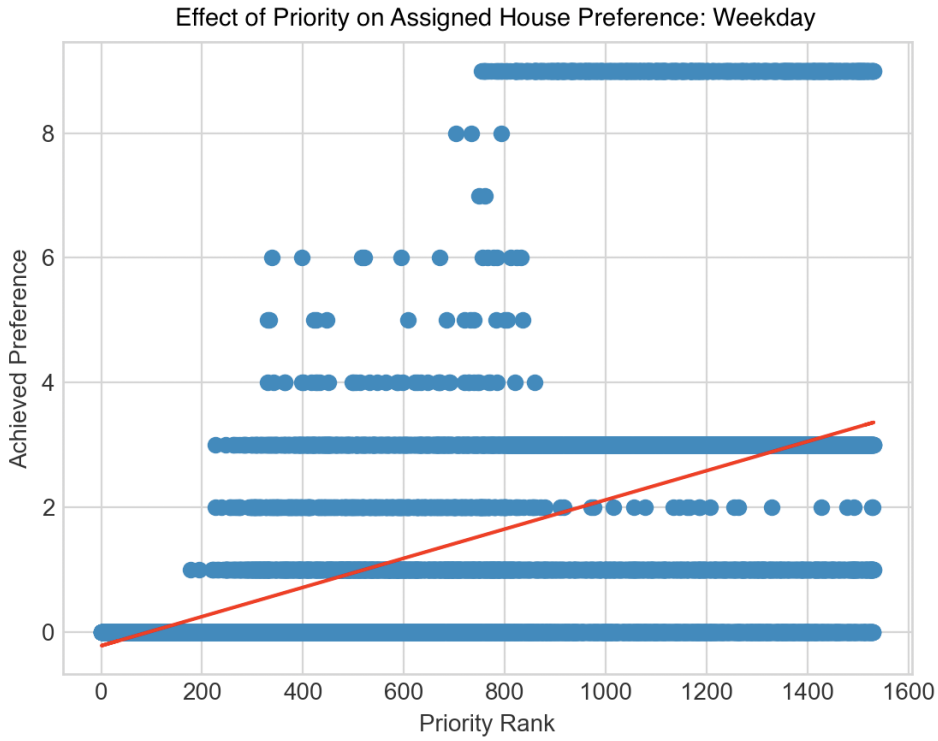
Average Improvement	Average Number of Swaps	Average Cycle Length
0.810	1146.6	3.563



3.2.5 Priority Analysis

Here we measure the impacts of introducing the priority concept on TTC. Since priority is randomized and it is crucial for being able to trade because it determines the order of trades, students who receive a high priority relative to their housemates will have

a massive advantage over those at the bottom of the priority rank of their house. We can visualize the effect in the following plots showing long run average outcomes of weekday and weekend dinner simulations:



We can see here that on weekdays when agents express more indifference and less particularity, expected achieved preference moves from 0 (top preference) to about 3.5 (between fourth and fifth preference) across the priority ordering

On weekends, when agents are more particular, it seems that achieved preference remains very strong across most priority ranks.

3.3 Discussion

We notice that the average improvement for weekday lunch is the worst among all four meal types and that weekday dinner follows closely behind. This can be explained by the larger amount of competition for specific houses during the weekdays. More people have as their first choice River Central houses but because there is a limited number of students to trade with, these students must fall back to their second, third and fourth choice more often. We can see this dynamic in the weekday lunch and dinner graphs as compared to the weekend lunch and dinner graphs. In the weekday lunch and dinner a significant number of students get their 1st, 2nd, 3rd and 4th choice while for the weekend most students get either their 1st or 2nd choice. One interesting fact of the graph is that very few agents get their third choice in weekday lunch and dinner while a significant number receive their fourth choice. At first this seems odd but by understanding the construction of the preference orderings it makes sense. River Central is composed of 3 houses and for each River Central agent they randomize their preferences across the 3 houses. This means that in the TTC most River Central agents will get their first choice, while some will get their second. However, by the time we get to their third choice it is very likely that the house has already been matched with other agents who put that house first or second in their preferences. This can also explain why the average improvement for weekday lunch and dinner is lower than that for weekend lunch and dinner.

Some other interesting results are that the average cycle length for the four meal times stayed about the same despite the variance in competition and number of total agents in the system. There does appear to be a slight increase in this value as we proceed from weekday meals to weekend meals and from lunch to dinner within days. Longer cycle lengths could be explained by more equal participation across the trades by all houses leading to fewer agents being stuck at their own house later down in the priority rankings. Additionally, during the testing of our implementation we created the priority ordering in two different ways. We randomized the order across all agents and we ran simulations where we randomized the ordering of the real agents but put all the dummy agents at the end of the priority ranking. These two problem instances did not actually change the result of the TTC mechanism because no dummy agents were pointing at a house that had dummy agents pointing back at them.

From the priority graphics, we can observe that the random priority assignment of an agent in an instance of the TTC mechanism has a measurable impact on the outcome of that agent. The trend across both weekday and weekend scenarios is that a high priority leads to a high probability of achieving one of your top preferences for dining hall assignment. One interesting thing to note is how certain achieved preferences are more common than others across all priorities which is an artifact of how we defined the different agent strategies. Agents generally had either one, three, or nine preferences greater than their initial house which is why the outcomes tend to clump around top, second, fourth, and tenth preferences. Following the provided descriptive statistics from 3.2.1-4, the average improvement for the weekdays was lower than for weekends. This can be seen in these graphs by the steeper line in the weekday graph. This steepness indicates that on weekends when agents had more particular and strict preferences, the mechanism was successful at getting agents into their top choice. In fact, on weekends, even the lowest priority agent is predicted by the linear model to receive their first or second choice. Weekdays have more agents who have larger priority orders over different houses and are less particular. With larger preference orders, an agents own house is lower in their preference order so agents are more likely to end up with a lower preference so it appears that the mechanism has a lower average achieved preferences for agents. However, most agents are still getting within their top three choices, even for agents with a low ranking. So while priority ranking is impactful on assignment, we do not think that it creates a large imbalance in outcome and it does not break the strategy proof feature of the TTC mechanism.

4 Conclusions

4.1 Takeaways

One of the most important takeaways from our work on this project was the process and lessons learned while implementing a full mechanism from scratch. It involved thinking through code design as well as collaborating effectively within the group to produce well structured code. We also recognized the value in thinking about real world problems that could be solved with the mechanisms we learned in class or adapted versions of those mechanisms. Clearly, the material learned in CS 136 is easily adaptable to real world problems.

4.2 Looking Forward

Our implementation of our adapted TTC mechanism has more simulation potential that we did not have the time to test. With the ability to create agents from 12 houses with different strategies and preference profiles, the combinations of simulation environments is endless. Tinkering with the rules surrounding creating unique Dummy agents could allow for some surprisingly complex expressions of interhouse dining rules as we have shown in this paper.

We believe that the mechanism has good potential to be adopted by HUDS to replace or supplement its current system of interhouse restrictions. We could begin by reaching out to Harvard College administration to explain to them the interest and benefits of implementing such a mechanism. We could then develop a mobile application running our modified TTC Mechanism which could be incorporated into the Harvard College Mobile App. The app could provide real data and demographics as well as related data of people interacting in the system (class enrollment, year, concentration, etc.). We could also get even better insight into how to distribute food resources across dining halls. Since the mechanism is strategy proof, as long as we communicate to the student body that truthful reporting is a dominant strategy equilibrium, the data collected will be accurate. The mechanism would reduce the friction between students mixing between houses and allow the dining staff better control and forecast over future meals.