Introduction.

We tackle the problem of maintaining diversity during the process of Interhouse Transfer, which allows students to move out of the house they were assigned after freshman year. The current allocation mechanism, based on randomized serial dictatorship, does a poor job in preserving diversity, especially when students of a similar type share similar preferences. Our goal is to create a mechanism for Interhouse Transfer based around the top trading cycle algorithm that maintains desirable properties like strategy proofness and high transfer rate, while also preserving diversity, as well as devise a method to analyze both qualitatively and quantitatively the improvement.

Model Overview.

Our model consists of a class called `Block`**, which represents a group of students entering interhouse transfer, and an adjusted TTC interhouse mechanism that allocates** `Block`**s to houses. A** `Block` **object has a type, which determines its preference order for the twelve Harvard houses, and a randomly-assigned current house. Our interhouse mechanism adds noise to the Block object preference so that:**

1. **Participation satisfies individual rationality**
2. **Reporting truthful group size and preference order is a dominating strategy**
3. **The allocation of Blocks to houses maintains diversity across types**

Blocks.

A `Block` **object is initialized with three attributes:** `house`**,** `pref_type`**, and** `block_type`**. House stores the current house of the** `Block` **object.** `Pref_type` **stores how many options there are for the block_type (either 3 in** `"basic"` **or 6 in** `"permutation"`**), each of which is associated with a specific preference order.** `Block_type` **determines exactly which of the 3 or 6 options is the type of the block.**

To describe the preference order associated with a `block_type`**, we split the houses into three neighborhoods of four houses. Each** `block_type` **prefers the neighborhoods in some strict order (there are 6), and we assume that the preference order within neighborhoods is the same across all types. More explicitly, we have for any object** `block` **of class** `Block`**:**

```
block.nbhdA = [1, 2, 3, 4]
```

```
block.nbhdB = [5, 6, 7, 8]
block.nbhdC = [9, 10, 11, 12]
if block_type == 1 && pref_type == "permutation":
     block.pref = block.nbhdA + block.nbhdB + block.nbhdC
```

We express preferences (`block.pref`) as a list of houses from greatest to least preference, where `block.pref` is a permutation of A, B, C. Consequently all block types prefer houses at the beginning of a neighborhood over houses at the end of the neighborhood. Later on, we add noise to `block.pref` so different blocks of the same type have different, yet correlated preferences.

Assumptions in block_types.

Permuting the neighborhoods assumes that Harvard students of different "types" will prefer to be in different neighborhoods of Harvard, and do not universally have the same preference order for houses. (Note that if Harvard students had homogenous housing preferences, then interhouse would be an uninteresting problem: no one in the top house would want to move out, so then no one in the second best house would want to move out, and so on!) More concretely, permuting the neighborhoods describes preferences on social/locational aspects, which are generally higher priority for students than basic housing quality. As seen with the common argument against the Quad, despite the Quad having the best housing options as measured by rooms per person, and notoriously good dining hall food, Harvard students prioritize social/locational quality over housing facilities quality. Nonetheless, some students strongly prefer the Quad due to its proximity to Maxwell Dworkin, its housing options, and its higher housing satisfaction rate, so the Quad is not universally disliked. We group the houses by social/locational quality through neighborhoods, and exhibit different priorities by permuting the neighborhoods to create a preference order.

In block_types, we keep the houses in the same order within a neighborhood to reflect that, once social/locational preferences are satisfied, Harvard students prefer better housing facilities over worse housing facilities, and that the judgement of which housing facilities are better within a neighborhood is similar across types. This is evidenced by common housing complaints and preferences in a few houses across Harvard: Adams has

too dim lighting, Kirkland and Eliot are dark and unclean, whereas Quincy is considered one of the best river houses.

Implementing block_types.

In our model, every student belongs to a type, because types are supposed to reflect mainstream preferences. However, it is unlikely that students of the same type have exactly the same preference order. To imitate this effect, for each `Block` **object we add noise to the preference order described by** `block_type` **to get correlated, but unidentical preferences across** `Blocks` **of the same type. Concretely, if** `block_type = 1` **gave a preference of:**

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12],
```

then an individual block of `block_type = 1, house = 12`**, might have preference order of:**

```
[1, 2, 4, 5, 3, 7, 6, 8, 10, 12, 12, 12].
```

Note that the `house` **represented by** `12` **is repeated so the preference is IR: a block never has houses in its preference order that are worse than its preference order.**

In adding noise for the block-specific preference, we wanted to permute houses between neighborhoods in the block_type while avoiding large movements. The moderate noise reflects that student across types might disagree slightly on the tradeoff between social/locational and housing facilities. While we might want the third-priority house in the `block_type` **preference to appear in positions 1-5 of the noisy** `block.pref` **, we would not want it in the bottom neighborhood priority in** `block.pref`. **To model this noise, we assign the houses in the** `block_type` **preference logistical weights (Figure 1) according to their priority, and then repeatedly randomly select from the weighted houses to create** `block.pref`. **(We chose a logistical curve rather than a normal CDF because the logistic distribution has higher kurtosis than the normal distribution.) Because we have to normalize the logistic weights, the outcome of adding noise was more desirable when we assigned lower weights to higher priority houses, and then reversed the randomly generated preference. The function used for assigning weights describes sampling 12 points evenly between -7 and 7 on a logistic curve with steepness 1. For place 0 < x < 13 in** `block_type` **preference:**

$$weight(x) = \frac{1}{1 + e^{-(-\frac{14}{13}x + 7)}}$$

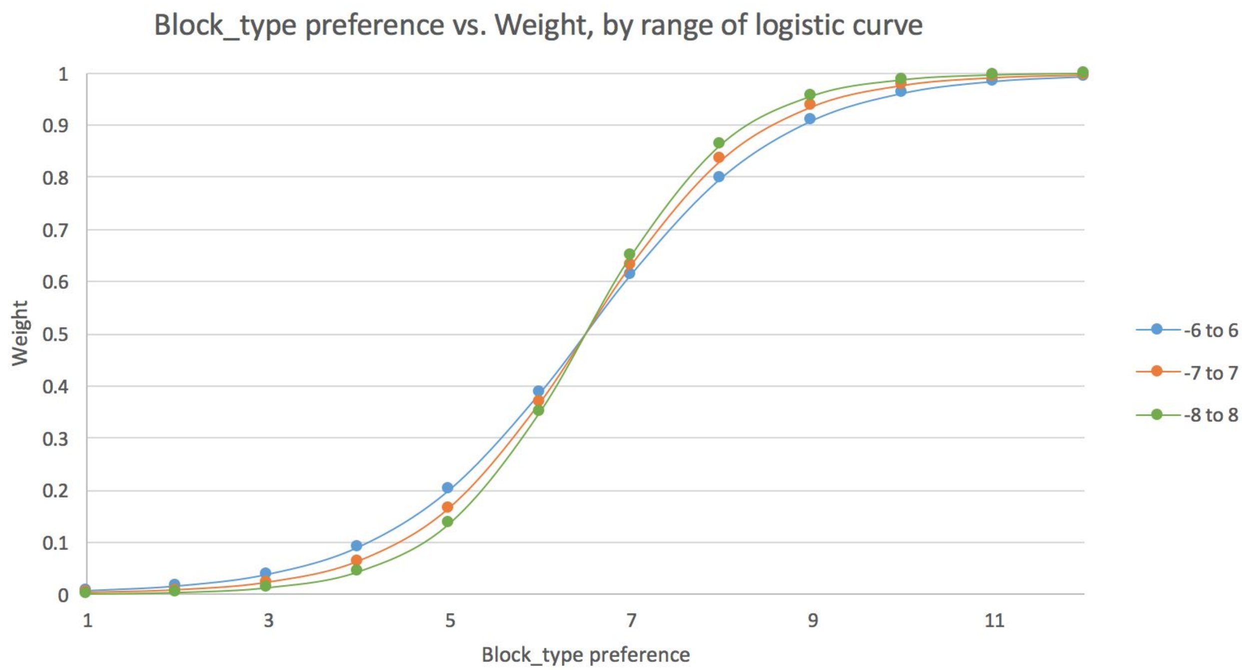


Figure 1: Visualization of weights assigned to `block_type` preferences for different ranges on the logistic curve.

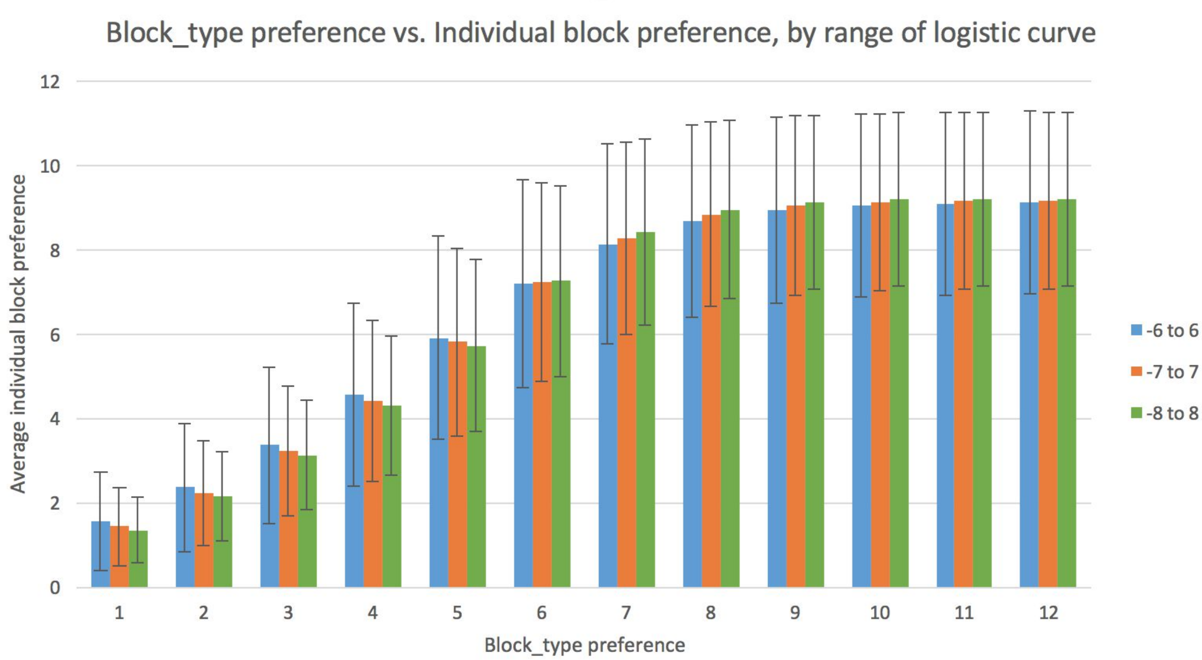


Figure 2: Visualization of average individual block preferences vs. `block_type` preference for different ranges on the logistic curve.

Analysis of adding logistical noise to `block_type` preferences.

Explicitly, the process of assigning preferences to a certain block is to determine the block_type, which gives a preference list prototype, and apply noise to this preference list. In adding noise, we:

1. **Sample 12 points evenly from the logistic curve with steepness -1 on range (-7, 7)**
2. **Assign lower weights to more-preferred houses**
3. **Randomly select from the weighted houses (heavier houses, or less-preferred houses, are more likely to be selected first)**
4. **Reverse the resulting list so more-preferred houses are at the top of the preference list**

From Figure 1, sampling noise from a smaller range (-6 to 6) gives higher weights to houses within the top half of `block_type` preferences, as well as lower weights the bottom half of `block_type` preferences. Note that the weight of the fourth highest preferred house is almost double in range (-6, 6) than in (-8, 8). Consequently, the blocking preferences generated by weighting in range (-6, 6) are on average farther from the `block_type` preference and noiser among themselves than the blocking preferences generated by weighting in range (-8, 8), as seen in Figure 2. When weighting in range (-6, 6), top-preferred houses appear lower in preference in `block.pref` and least-preferred houses appear higher in preference in `block.pref`. Meanwhile, weighting in range (-8, 8) gives top-preferred houses higher preference, least-preferred houses less preference, and produces less noise among the resulting preferences.

Although weighting in a larger range produces less noisy preferences, the high standard deviation means that individual preferences weighted in range (-6, 6), (-7, 7), and (-8, 8) do not appear significantly different. For example, Table 1 demonstrates three preference lists generated with each type of weighting. The choice of weighting (-7, 7) simply picked the middle of three reasonable choices.

Range	block.pref, example 1	block.pref, example 2	block.pref, example 3
(-6, 6)	[1, 3, 2, 5, 10, 4, 9, 6, 11, 7, 8, 12]	[2, 1, 3, 4, 6, 9, 12, 8, 5, 10, 7, 11]	[3, 2, 4, 1, 7, 5, 10, 9, 6, 8, 11, 12]
(-7, 7)	[1, 2, 4, 3, 5, 6, 11, 7, 10, 9, 8, 12]	[2, 3, 1, 4, 10, 12, 7, 8, 6, 11, 5, 9]	[1, 3, 2, 5, 7, 4, 10, 8, 11, 6, 12, 9]
(-8, 8)	[1, 3, 4, 2, 5, 12, 7, 8, 9, 6, 11, 10]	[1, 2, 3, 4, 6, 5, 10, 8, 7, 9, 12, 11]	[2, 3, 1, 5, 9, 7, 10, 8, 4, 11, 6, 12]

Table 1: Three blocking preferences generated from block_type [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12] with differing weights.

Interhouse TTC Mechanism.

Our interhouse mechanism uses TTC among student groups (blocks) that have the same number of students. However, TTC requires strict preference orders on student groups (specifically: would you rather trade with student group A in Mather, or student group B?) even though students report only their preference order on the houses. To generate strict preference from students' preference order on the houses, we randomly order student blocks within a house as s_1, s_2, s_3, …, s_n such that all student groups in TTC prefer s_1 over s_2 over s_3 over s_n. This is similar to an RSD because a student block s_k in a house cannot trade until s_{k-1} has traded.

The use of RSD in trading means that only one student per house has in-edges in each round of TTC. Consequently all trades in each round of TTC happen between the top student in each house, and we only have to keep track of 12 students (one per house) in each round of TTC among student groups with the same number of students. Furthermore, because changing preferences does not change the priority of student blocks, and TTC is strategy-proof, our interhouse TTC mechanism has truthful house-preference reporting as a dominant strategy.

It is difficult to make the interhouse TTC mechanism completely strategy-proof in terms of truthfully reporting group size, because if there is only one student group of size 8, that group of 8 has a 0 chance of trading with another group of 8 for a better house. However, we can discourage reporting multiple smaller groups if all size groups have approximately the same probability (< 1) at transferring to a better house, and discourage reporting larger groups by adding larger amounts of logistic noise to the submitted preferences of larger groups. Note that adding noise does not change the property that

reporting truthful housing preferences is a dominating strategy, and also helps preserve diversity.

Penalizing large groups: pick who you live with, or where you live.

To preserve diversity, we will penalize groups entering TTC by adding logistic noise to their reported preference order but keeping the result IR. Larger groups will receive more noise, because they are likely students of the same type and detract more from diversity in the house. Our model for logistic noise takes the same form as noise in `Block`, except logistic noise is sampled, for groups of x students, from the range:

$$(-0.1 - 0.01x, 0.1 - 0.01x)$$

As mentioned before, a larger range has less noise, so larger groups receive more noise. Because the range is so small, every group in TTC receives a lot of noise, and transfer rates are lowered. Further research into interhouse transfer could determine the actual distribution of types by asking all students to report their housing preferences and trying to cluster the preferences, and this likely would allow for less noise. However, preserving diversity will require removing almost all identifiers of each student type in a preference order.

In Figure 3, we see that increasing group size adds a slight amount of noise, so groups are not motivated to join and form larger groups. However, the group-related penalty is dwarfed by the penalty established by entering TTC: in the results section, we will demonstrate through Shannon entropy that this noise produces reasonable diversity among the houses.

Note that despite the amount of noise added by TTC, simulations that assume uniform distribution across student types have transfer rates of up to 45% while preserving diversity. More will be discussed in the results section.

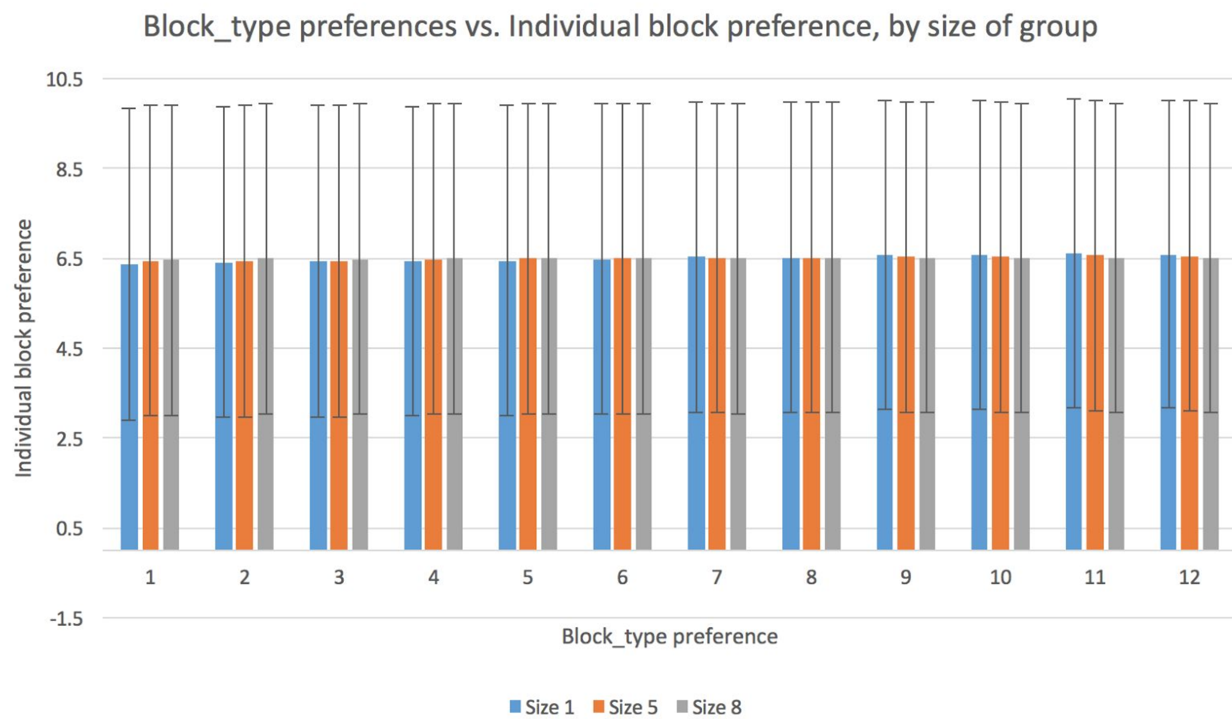


Figure 3: Visualization of average individual block preferences vs. `block_type` **preference for group sizes**

Limited transfer rate discourages separation into multiple smaller groups.

Because we penalize larger groups by a negligible amount, the transfer rate between differently-sized groups differs by less than 1% if the same number of blocks of each size enters TTC. We can define a function *f(x)* **that maps the number of blocks of any size to the anticipated transfer rate. (See Figure 3.) Observe that** *f(x) << f(x)^2*, **and** *f(8) > f(1000)^2*. **Because** *f(x)* **is the probability that a given group transfers successfully out of its house , and** *f(x)^2* **is the probability that two given groups transfer successfully out of their houses,** *f(x) << f(x)^2* **implies that splitting into multiple subgroups is heavily disadvantageous. The probability that all of the subgroups transfer out successfully-- not even accounting for the additional stipulation that they must end up in the same house-- is lower than the probability of the larger group transferring out of the house. Even if there are only 8 groups of 8 and 1000 groups of 4, a group of 8 should not want to split.**

Note that if there is extremely limited liquidity, the inequality *f(1) < f(500)^2* **holds.**

However, it is unlikely that there is almost no liquidity in one group size, and extremely large liquidity in another size. Assigning large negative utility to exposure (one subgroup moves out, and another doesn't) or the likely outcome of subgroups transferring to different houses means that our mechanism discourages separation and is strategy-proof given any reasonable amount of participation in Interhouse.

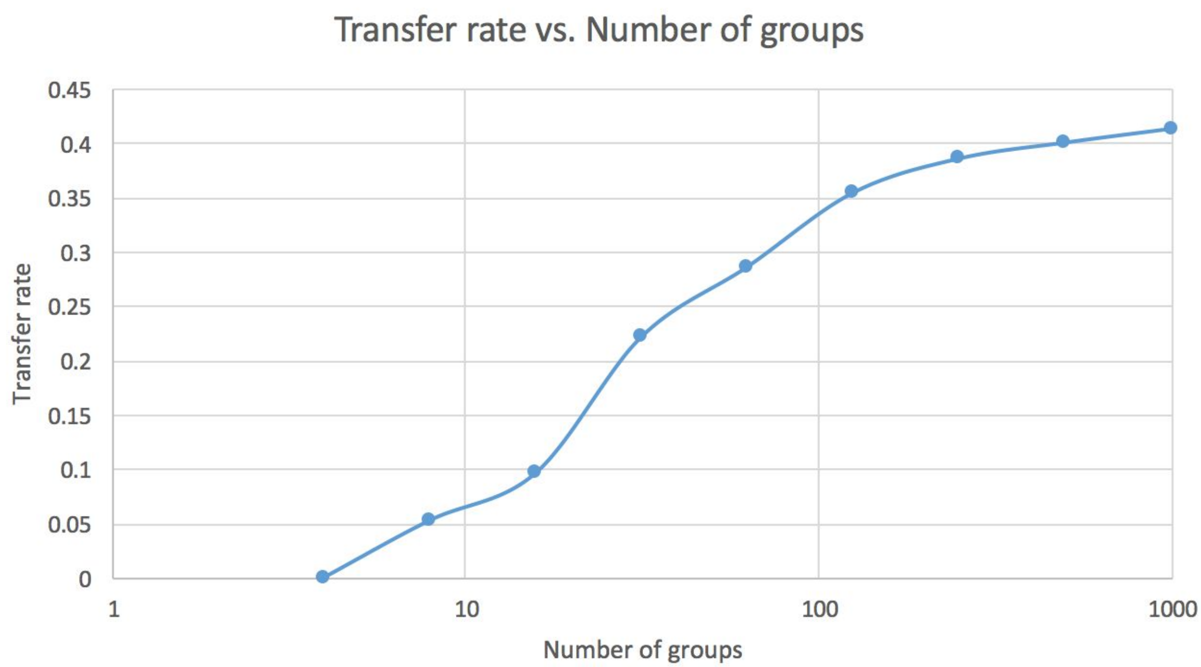


Figure 4: Transfer rate among groups of the same size, plotted against group size. The x-axis is scaled logarithmically.

Empirical Results.

We now assess the success of the implemented TTC - incorporating mechanism in achieving our desired objective of increased diversity when compared to that of the current RSD mechanism. We also test the robustness of these results against a variety of parameters. We make sure that any significant changes in diversity are consistently seen across a large range of reasonable assumptions about the proportions of each type in the population and the initial distributions of those types across all the houses. We achieve a quantitative sense of diversity by calculating the entropy of the allocation returned by a particular mechanism, and a qualitative sense by depicting the allocation in a bar graph.

RSD mechanism:

We first run the RSD mechanism on a sample of 5000 students and a blocking group size of 8 students. The types for each of the 5000 students are distributed uniformly across 6 different options and, within each type, the initial house of each student is also distributed uniformly across 12 different options. We calculate a total entropy of 4.73169184609 in this instance, and we plot the resulting allocation in Figure 5. We can immediately see the problem, and the lack of diversity becomes highly apparent. Almost everyone is able to transfer to one of their most preferred houses (Note: this is the case because we decided to ignore the issue of individual rationality in the RSD mechanism in order to focus on the effects on diversity), but this comes at the cost of diversity. Students with overlapping preferences overwhelmingly end up in the same neighborhood to the exclusion of other types of students. We run the RSD mechanism on a second case, where the students are not uniformly distributed across type and house and instead have higher concentrations in a particular population or a particular house. We achieve this by drawing the proportions of types and proportion of initial houses for each type from a Dirichlet distribution, a distribution that allows to set concentration parameters around which the distribution will return a vector of values that add up to 1 and which are centered around the concentration parameters chosen. We then multiplied these results by the total number of students to get

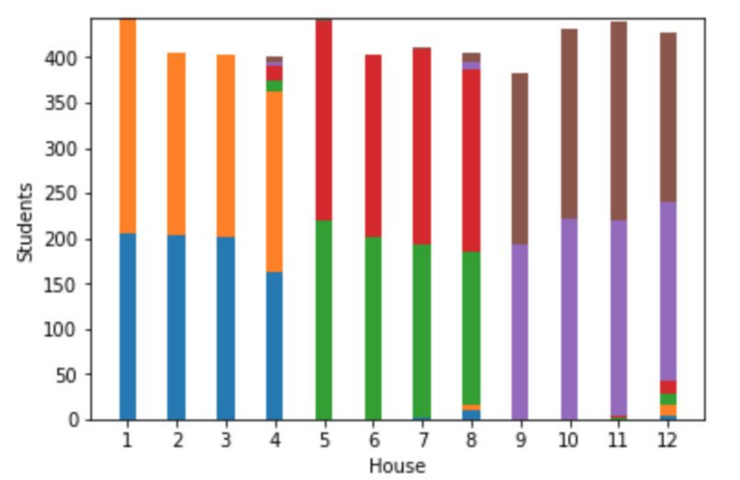


Figure 5: RSD allocation across uniform types and initial housing

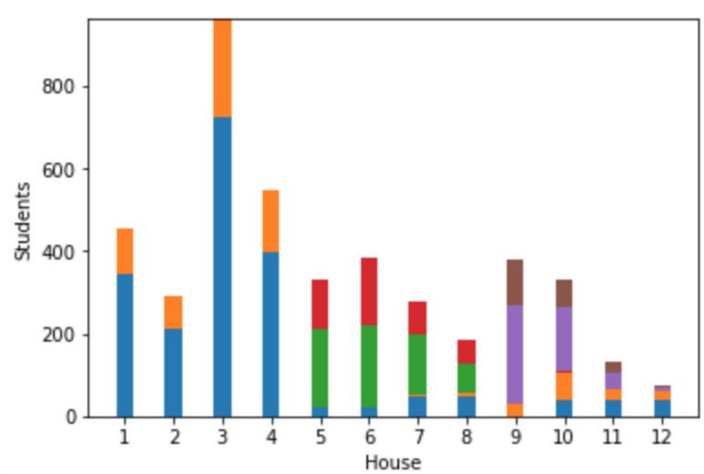
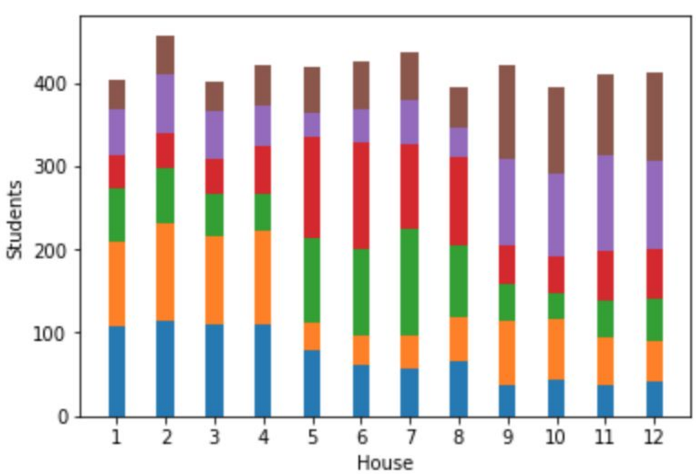


Figure 6: RSD allocation across concentrated types and initial housing

the desired proportion of students and the number of students of a particular type for each type to get the desired initial distribution of types across houses. We made sure to choose a random permutation of the proportions when determining initial housing in order to prevent the highest concentration of each type all coinciding in the same house initially. The parameters we chose for this second run were a ratio of 12 : 4 : 3 : 2 : 2 : 1 across the different types and a ratio of 8 : 4 : 2 : 2 : 2 : 2 : 1 : 1 : 1 : 1 : 1 : 1 for initial assignment to houses within each type. We observe that the overall diversity in these circumstances is even worse than in the uniform case, with a total entropy of 4.49318017786 and a final allocation as shown in Figure 6.

TTC - based mechanism:

When we run the TTC - based mechanism on the same parameters as above, we can see drastic improvements in the overall diversity after the final allocation. Again, with a population of 5000 students in groups of 8, on 6 types and 12 houses, and uniform distribution, we calculate a total entropy of 6.0406109545 in this case, significantly greater than the value of 4.73169184609 we received in the first RSD allocation. We plot the resulting TTC allocation in Figure 7. We can see a far more equitable arrangement of the members of each type across houses compared to with the RSD allocation in Figure 5. While we still see that each house seems to maintain the property that a couple of types account for a slim majority of the house, it is nowhere near as disparate as in the RSD allocation, where we see a couple of types dominating the whole house to the exclusion of other types. When applied to the uneven distribution of types and initial



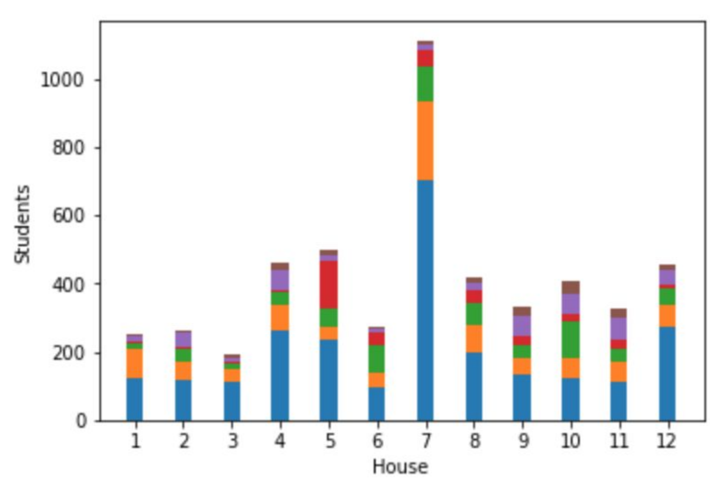


Figure 7: TTC allocation across uniform types and initial housing

Figure 8: TTC allocation across concentrated types and initial housing

housing with the same ratios as the second RSD
run from above, we also see a far greater proliferation of diversity. The total entropy we calculate in the second TTC case is 5.36202709881, also significantly greater than the value of 4.49318017786 derived in the second RSD allocation. The second TTC allocation is depicted in Figure 8. Where we saw types limited to the neighborhoods they most preferred in the RSD allocation we now see that types are far more evenly distributed across every house in the TTC allocation.

Splitting Strategy.

We also ran a quick empirical analysis on whether it would be a useful strategy to split one's larger group into two smaller groups. The desire would be to end up transferring to the same preferred house with a greater chance than if they had not split. We tested this for a specific example, imagining a group of 8 that might want to split into two groups of 4. We then generated a uniform initial housing distribution with types being distributed uniformly and houses within types being distributed uniformly as well, for 5000 students and group sizes of 4. However, then we added two more groups of 4, each with the same initial house and exactly the same preferences and tagged them with an identifier. We then ran the mechanism and saw whether or not the two groups ended up both transferring out of their old house to the same new house. We ran this 100 times and observed that this only happened 15 times, for a success rate of 15%. Considering that if the two groups had remained one group of 8 they would have been guaranteed to end up in the same house and the transfer rate observed with these group sizes was around 40% - 45%, we

concluded, at least for this scenario, splitting into smaller subgroups would not be a useful strategy. The noise added by the TTC mechanism and its emphasis on diversity has the added bonus of preventing untruthful reports on group size like this, as even when the groups report exactly the same preferences they can often end up in different houses.

Conclusion.

The issue of maintaining diversity during Interhouse Transfer to help foster a more balanced living experience for students is an important yet subtle one. We were able to uncovering the shortcomings of the current RSD mechanism by employing a quantitative measure of the diversity of an allocation using an analogy to entropy and a qualitative analysis by generating visualizations of the allocations with bar graphs. We then constructed a new mechanism centered around the TTC algorithm to prevent the loss of diversity after Interhouse Transfer, achieving this goal by injecting the right amount of randomness into the mechanism in order to prevent the polarization of agents with similar preference orders. This was done through the addition of logistical noise to the reported preferences of agents based on group size. We then demonstrated how these variations maintained the desirable properties of the TTC algorithm, from strategy-proofness to individual rationality, while drastically improving the diversity of the final allocation, evaluating the results with the quantitative and qualitative measures mentioned previously and comparing them to those returned by the RSD mechanism. In the future, we could look for ways to improve the transfer rate achieved by the TTC mechanism and perhaps ways to allow for groups of different sizes to enter the same market.