Performance of School Matching Algorithms

Omar Hossain Ahsan
Advised by Professor Peter Cramton
University of Maryland-College Park
oahsan@umd.edu

Abstract

A school match, or lottery, is a system in which students are matched to schools through a process in which they list their most preferred schools and are then matched to one of those schools using an algorithm. Two algorithms that have been studied in the past are a Deferred Acceptance model used in New York [1] and a Priority Matching mechanism used in Boston [2]. In this paper we study the performance of these school matching algorithms in Washington, D.C. and the factors that influence their success. In particular, we focus on the number of preferences listed by students as well as different ways they put relative importance on the academic quality of schools and their distance from the schools. We create a program that generates student data and preferences based on Washington D.C.'s population distribution. The schools in the program are identical to the ones that participated in the D.C. school match. The program creates a score for each school for each student that then is used to generate the preferences for every student. Since we generate preferences based on cardinal scores, we have information on not just which schools a student prefers, but how much they prefer each school over the next best one. We study the performance of the two algorithms by comparing the distribution of student utilities as the number of choices listed varies. We find that the New York algorithm achieves higher student utility on average than the Boston algorithm when students list a small number of schools in their preferences and that the Boston algorithm performs better when they list a larger number of schools.

1 Introduction

In a traditional school system, students are assigned to schools based on where they live. Every school has a designated zone, which is roughly the neighborhoods for which that school is the nearest school. Students are then required to attend the school whose zone they reside in. One issue with this system is that it does not allow students any possibility of choosing their own schools if their address is taken as fixed. To some degree this system does allow for school choice if people are free to live in any part of the district they choose. Then parents could simply choose to live in a neighborhood that is assigned to the school they want their children to attend . However, it is generally only wealthy people that have the freedom to live in any part of the county that they choose. The quality of the schools is usually public information, which leads to the corresponding neighborhoods having higher housing prices.

Because traditional school systems only allow the richer families to have a choice in the schools they send their kids to, some school systems have changed to school matches. A school match (or lottery) is a system in which students are assigned to schools based on their priorities at the schools and their own preferences. A student's priority at a school can be thought of as the rank the school assigns that student. In this paper we take the priorities of the students at each school as randomly determined. We assume that students will decide which schools they prefer by their academic quality and the schools' distance from their home. Students are then matched to the schools using a matching algorithm that uses the above information.

Two examples of matching algorithms are a Deferred Acceptance algorithm, which is in use in New York City [1], and a Priority Matching algorithm, which is used in Boston [2]. The two algorithms differ in that the Deferred Acceptance algorithm only issues one acceptance by a school to each student. It prioritizes the students that have a higher priority number at each school. It allows for students who matched with a school early (so, as one of their higher choices) to be displaced by another student who preferred the school less but had a higher priority at it. This algorithm is "strategy-proof," meaning that the dominant strategy for students is to list their true preferences instead of deciding to list a school they like less but think they are more likely to receive near the top as a "safety school." The Boston algorithm is different from this in that students may receive multiple offers from different schools. Additionally, students who rank a school higher on their preferences cannot be displaced by other students who had higher priorities but ranked the school lower. Unlike the Deferred Acceptance algorithm, this algorithm is not strategy proof. Students are advised by Boston Public Schools to consider ranking a less popular school near the top of their preferences in order to better their chances of getting a match.

In this paper we examine the performance of these school matching algorithms in Washington D.C. We evaluate the performance of these algorithms by comparing the utility of each student when matched to a school using each algorithm. In 2014, Washington held its first public school match using an algorithm similar to the Deferred Acceptance algorithm in New York. 71 percent of students were matched to one of their choices in the first year.

1.1 Literature Review

The merits of a school system with school choice over a school system where students do not get a choice in the school they attend are studied in two papers by Dr. Caroline Hoxby. Hoxby argues that school choice will lead to net increases in school productivity since less productive schools will see their students begin to leave for the more productive schools since they are able to have some degree of choice in their enrollments [4]. Hoxby cites three examples in Milwaukee, Michigan and Arizona where the competition posed by charter schools which allowed students to choose other schools lead to a boost in the productivity seen in public schools. The theoretical base for this increase in productivity in public schools is explained as the public schools get their funding from local property taxes. If parents are able to choose between school districts they will favor ones that are more productive at equivalent levels of property taxes. Therefore if the productivity in the public schools in an area are low they will also see a loss in funding over time as parents choose not to live in the area.

Hoxby also studies the benefits to student achievement, though first arguing that looking at student achievement is wrong-headed because studies are usually not able to study student achievement while holding school resources constant. To study gains in achievement, Hoxby looks at the specific example of Edison schools. Edison is a for-profit schooling company that is often hired to turn around failing schools, often in very low income areas. Hoxby uses test score data and finds that Edison students are 2.1 percent more likely to be judged proficient on their states exams when compared to their best match from public schools in the same district. Hoxby finds the best match by using a measure called the selected average treatment effect from another paper to find which public school was closest to the Edison school in this measure. The increase in achievement by attending these Edison schools provides some evidence that may suggest that giving more students access to schools like Edison by implementing school choice mechanisms may result in increases in student achievement [5].

Matching algorithms also have applications outside of matching students to high schools. Another matching mechanism can be found in matching recent medical school graduates to residency programs [6]. The paper talks about how early matching mechanisms in matching students to residencies in the United Kingdom suffered from stability issues. That is, situations where a student and a hospital would both prefer to be matched to each other rather than their current partners as produced by the matching. The question was asked if there could be an algorithm that not only produced a stable matching, but one that would make the dominant strategy of people participating in the match the list their true preferences, without any sort of safety schools. [6]

2 Methodology

In order to study the performance of these matching algorithms in Washington, D.C., we create a computer program to generate a dataset based on the characteristics of Washington D.C. We then run the two matching algorithms on the district to compare the performance of the two algorithms. In the Deferred Acceptance algorithm (used in New York and Washington D.C.), students are matched to schools as follows [1][3]:

Deferred Acceptance Algorithm (New York Algorithm)
1. Each student applies to their top choice amongst the schools they have not been rejected from. This means that if in the previous round of the algorithm the student was not rejected from the last school they applied to, they do not apply to a different school in this round.
2. Schools consider the set of students that have applied and haven't been rejected. The students are ordered by priority number. The school holds the top n students where n is the capacity of the school. If more than n students have applied in this round then the rest of the students are issued rejections.
3. Steps 1 and 2 repeat until either all students are matched to a school or there are no schools left for any student.

In the Priority Matching algorithm, students are matched using the following steps [2]:

Priority Matching Algorithm (Boston Algorithm)
1. Each student applies to their i^{th} choice school in round i, where i starts at 1.
2. Schools consider the students who have applied in this round. The students are ordered by priority number and are assigned to the remaining seats in order until there are no applicants or seats left. Any remaining applicants are rejected.
3. Steps 1 and 2 repeat until there are no schools left for any student.

One key difference is that in the Boston algorithm, acceptances are permanent and students in later rounds cannot displace previously accepted students even if the later round student has a higher priority number.

2.1 Creating The Washington, D.C. Dataset

To create a dataset to represent Washington, D.C., we look at census tract data from the 2010 and a map of the census tracts. The key pieces of information we use from each district are its population, the percentage of the district that is school-age children, latitude and longitude. We then use this information to generate the appropriate number of students for each census tract area, locating them all at the census tracts given latitude and longitude. The number of students for each district is also impacted by the fraction of students in the tract that choose to attend private school instead of D.C. Public Schools (DCPS). In order to create the schools, we take the capacities and locations of each school as given by their capacities specified by DCPS and their locations given by their latitudes and longitudes.

2.2 Generating Student Preferences

In this model students generate preferences on schools based on the rankings and the locations of the schools. Students create an academic score for each school based on the schools performance on a countywide exam. Specifically for student i and school j, student i's academic score of school j is given by

$$\text{aScore}_{i,j} = \frac{\text{testScore}_j}{\max_k\{\text{testScore}_k\}}$$

for all schools k.

Students also create a distance score for each school which is based on the location of the school, the location of the student and the size of the county. For student i and school j, and length of the county given by D, student i's distance score of school j is given by

$$\text{dScore}_{i,j} = 1 - \frac{\|\text{location}_i - \text{location}_j\|}{D}$$

The distance score is subtracted from 1 in order to have shorter distances be mapped to higher scores. The student's final score is created by

$$\text{score}_{i,j} = \text{aScore}_{i,j} * \alpha + \text{dScore}_{i,j} * \beta + \epsilon$$

where $\alpha > 0,\ \beta > 0,\ \alpha + \beta = 1$. ϵ is a random term that is uniformly distributed between 0 and 1. The values of α and β are given in three different scenarios for the district. Scenario one is "typical" scenario, where students mostly care about the academic scores of the schools but also factor in distance score as well since they prefer shorter commutes. In this scenario they put 75% of the weight on academics ($\alpha = 0.75$) and 25% on distance ($\beta = 0.25$). Scenario two is used to represent a situation where the schools are all perceived as being of similar quality and allows for students to put additional value on distance with $\alpha = \beta = 0.5$. Scenario three is used to represent a situation where travel costs are largely negligble relative to the difference in academic scores of the schools and sets $\alpha = 0.9$ and $\beta = 0.1$.

In the New York model, students then generate their preferences simply based on the scores of each school i.e. if school i is scored higher than school j, then school i is ranked higher than school j. However in the Boston model, preferences cannot be generated this simply, as simply stating true preferences is not necesarily the best strategy for each student. Students in the Boston model will also take into account their perceived chances of getting into each school when generating their preferences. To generate the chances, we run the Boston algorithm four times and calculate the acceptance rates for the schools each time. We then run the algorithm one final time where students base their expectations on acceptance rates for each school based on data from the previous runs. Letting $x_{i,j}$ represent the acceptance rate for school i in year j we calculate

$$x_{i,j} = 0.2 * x_{i,j-3} + 0.3 * x_{i,j-2} + 0.5 * x_{i,j-1}$$

Using this information, for the Boston algorithm students generate their preferences based on the score of the school and the schools acceptance rate. For student a, school i is ranked before school j if

$$\text{score}_{a,i} * \text{acceptRate}_i \geq \text{score}_{a,j} * \text{acceptRate}_j$$

The matching algorithms are then evaluated based on the percentage of students that received their top choice and the percentage of students that do not receive any of their choices. Additionally we evaluate algorithms based on the average score of the school each student receives based on their preferences. We run these tests on the district representing Washington, D.C. intially studying the effect of the number of choices given by each student when using the first scenario ($\alpha = 0.75$, $\beta = 0.25$) and then also on the other scenarios.

3 Results

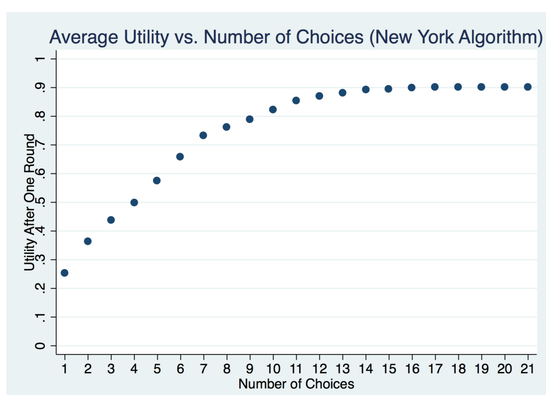
3.1 Effect of Number of Choices on Performance After One Round

In this section we study the performance of the New York and Boston algorithms on D.C. data after one round. After one round, not all students are guaranteed to be matched to a school. A student's utility is given by:

$$u_i = \begin{cases} \text{score}_{i,j} & \text{if student } i \text{ matched to school } j \\ 0 & \text{if student } i \text{ is unmatched} \end{cases}$$

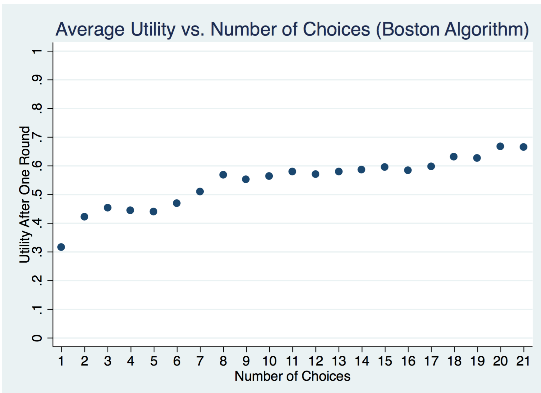
The simuaions run for section 3 are all based on the first scenario for α and β described above.

Figure 1: *Performance of New York algorithm*



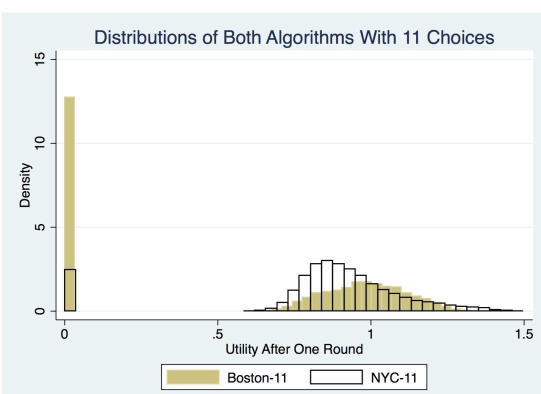
The New York algorithm sees large increases in performance when increasing the number of choices listed by each student. The marginal increase, however, decreases at high numbers of choices. This is not surprising as with a small number of choices, many students are being left unmatched. As students list more choices their odds of receiving one increase. Since students cannot be matched to more than one choice we also do not have some students taking up multiple spots so more students can be given a match.

Figure 2: *Performance of Boston algorithm*



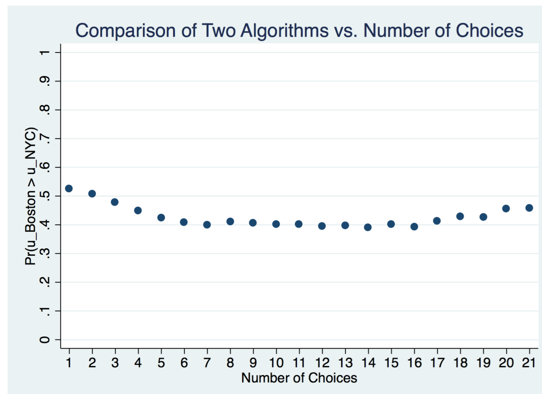
Increases in student utility as the number of choices increases are smaller in the Boston algorithm. One of the differences between the two is that the Boston algorithm allows students to be matched to multiple schools. This means that in the context of one round, many more students may be left unmatched as schools appear full. Though the schools will later be open for more applications the result is that after one round we see far more students left unmatched. This can be seen in the following histogram showing the distribution of student utilities when students are given 11 choices under both algorithms.

Figure 3: *Distributions of utilities under both algorithms with 11 choices per student*



As seen in the figure, the utilities amongst students that have been matched is often higher under the Boston algorithm than the New York algorithm. The New York algorithm, however, leaves far fewer students unmatched.

Figure 4: *Comparison of Boston and New York algorithms - the y-axis is the probability that a random utility from the Boston distribution of utilities is greater than a random utility from the New York distribution*

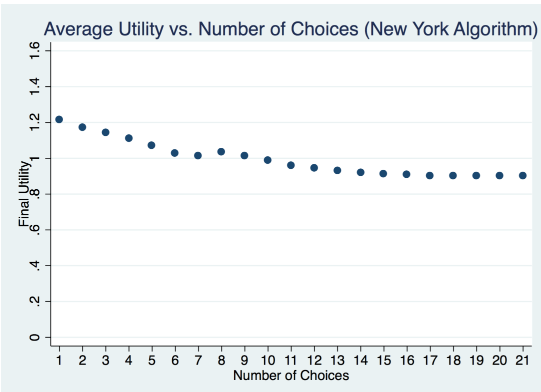


For a number of choices between 3 and 21 we observe that a random utility taken from the New York algorithm is expected to be higher than one taken from the Boston algorithm, Boston is expected to be better for number of choices equal to 1 or 2. However at both ends of the graph we find that the probability is close to 0.5 and that even in the center it remains relatively close to even. This contrasts from looking at the averages in the previous graphs where the New York algorithm's average utility is significantly higher at these points. This fits with the distribution seen in Figure 3 showing that while the Boston algorithm often performs better amongst matched students, it still leaves many unmatched after one round which is lowering the average utility of the students.

3.2 Effect of Number of Choices on Final Performance

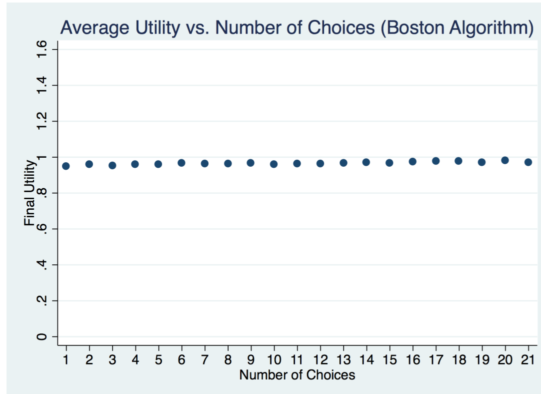
In this section we continue running the algorithms for multiple rounds until all students are matched to a school. After each iteration of the algorithm, students who were matched to a school are matched to that school (or, in the Boston algorithm simulation, the top school amongst the ones they were matched to) and are removed from the simulations. Any full schools are removed and the other schools have their capacities reduced appropriately. The simulations are then rerun with the remaining students.

Figure 5: *Performance of New York algorithm*



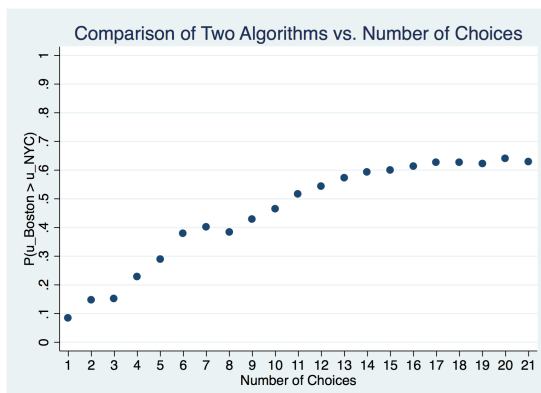
With the New York algorithm we see a different effect of increasing the number of choices on average student utility. When just looking at performance after one round we saw utility increase as the number of choices by each student given increased. This is largely due to far fewer students being left unmatched, rather than students getting more preferred choices. When looking at final performance the algorithm is re-run and no students are left unmatched regardless of the number of schools listed by each student. Thus this positive impact on the average utility is lost. However by increasing the number of schools on students' preferences we are actually increasing the probability that a student who listed a school as their top choice gets displaced by another student who liked the school less but had a higher priority number. In this case the decrease in utility may be explained by a student with higher preferences for the school being displaced by another student who actually does not prefer the school that much.

Figure 6: *Performance of Boston algorithm*



We see that the Boston algorithm's performance is relatively constant (compared to the changes seen in the New York algorithm) even as the number of schools allowed on students' preferences changes. There is just a small increase in average student utility as the number of choices increases.

Figure 7: *Comparison of Boston and New York algorithms*



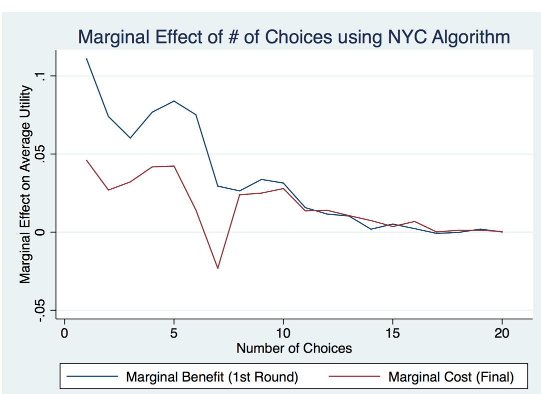
In this graph we see the trend between number of choices given by students and the probability that a random utility taken from the Boston distribution is higher than a random utility taken from the New York distribution. For number of choices between 1 and 10 we see that the New York algorithm achieves a better final result and that the Boston algorithm achieves a better final result for higher number of choices.

When deciding how many schools to allow students to list on their preferences we find that in the Boston algorithm the decision is rather simple. Increasing the number of choices is good because it improves the end result by a small amount and will lead to algorithm to finish faster. Increasing the number of choices beyond a certain point however may not be practical in real scenarios, as students may not thorough enough in ranking the schools to list that many choices at once.

For the New York algorithm districts should consider whether they want to have a better result after the first round, or have a better final result, as a smaller number will lead to a better final result but a worse first round. Although it might seem better to maximize the final utility for all students after the completion of the matching program, there may be justification for maximizing the utility after round 1 as well. Even if the end result is better, students may not be satisfied if a large number of them are told that they were unmatched after the first round and will have to go through the lottery again with fewer available spots in schools remaining.

In an effort to find an "optimal" number of choices for students when using the New York algorithm, we look at the marginal effect of increasing the number of choices on both the performance after one round and the final performance after all students are matched.

Figure 8: *Effect of increasing number of choices on performance of the New York algorithm*



The marginal benefit and marginal cost curves first intersect between 11 and 12 choices. Beyond that the cost curve is mostly above the benefit curve but they intersect again and the difference is small. For the D.C. district it is then good to have students list at least 11 choices because for numbers of choices between 1 and 10 the marginal cost for the end result is significantly below the marginal benefit for the result after one round. At low numbers of choices the benefits to increasing the number of choices outweigh the costs.

4 Simulations With Different Weights for Academics and Distance

As in the previous simulations, students score schools by computing:

$$\text{score}_{i,j} = \text{aScore}_{i,j} * \alpha + \text{dScore}_{i,j} * \beta + \epsilon$$

In this section we study how the performance of the two algorithms changes when the weights for academic quality and distance change. The above figures were all generated with the assumption that students weigh academic quality 75% ($\alpha = 0.75$) and distance 25% ($\beta = 0.25$). Here we consider two alternatives, $\alpha = 0.5$ (scenario two) and $\alpha = 0.9$ (scenario three). For these sections we always run simulations to completion and analyze performance after all students have been matched.

4.1 Scenario Two

In this section we consider a scenario where students perceive the academic quality of the schools as relatively similar and do not care so much about how much better one school is academically than another. So we increase β while maintaining $\alpha + \beta = 1$ and use the values $\alpha = \beta = 0.5$. Students here weigh the distance to schools and academics equally when making their choices.

Figure 9: *Performance of New York algorithm*

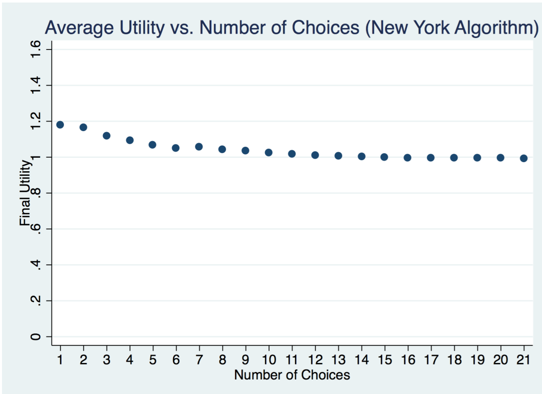
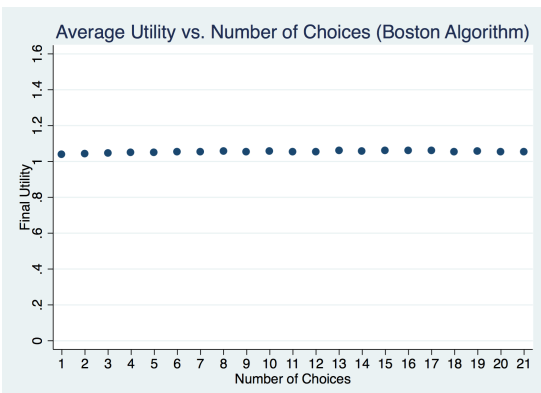
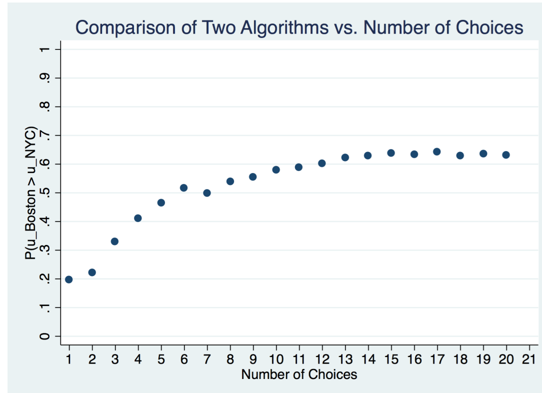


Figure 10: *Performance of Boston algorithm*



We observe the same relationship between number of choices and average utility for both algorithms that we did in scenario one. The main difference is that utilities are overall higher in scenario two. This is not surprising as if students are more likely to prefer schools closer to them then students' preferences will be more varied and will lead to more students being given their top choices.

Figure 11: *Comparison of Boston and New York algorithms*



Overall we see the same trend as in scenario one - that a random utility from the Boston algorithm is more likely to be higher than a random utility from the New York algorithm as the number of choices listed goes up. However, compared to scenario one, we see that this probability increases faster at first. The Boston algorithm performs better for all number of choices greater than 7 (and also at 5) compared to all number of choices greater than 11 in scenario one.

4.2 Scenario Three

In this section we consider a scenario where students consider travel cost to be relatively negligible relative to the differences between schools academically. A scenario for this might be a highly dense county that subsidizes transportation costs for students. Here we take $\alpha = 0.9$ and $\beta = 0.1$. Students consider academics to be far more important than distance when making preferences in this scenario.

Figure 12: *Performance of New York algorithm*

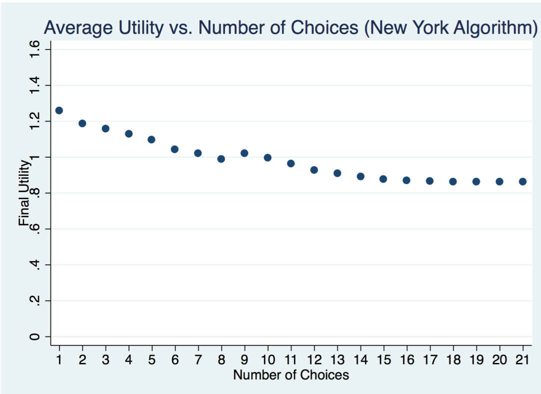
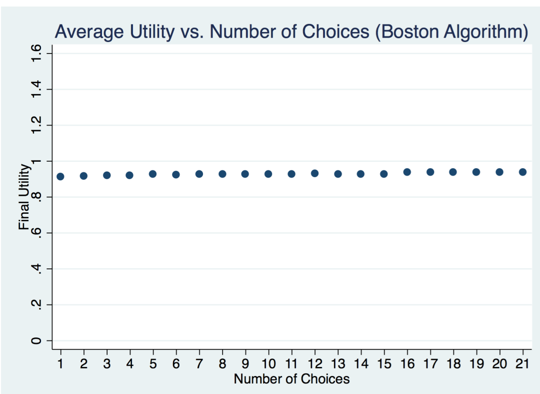
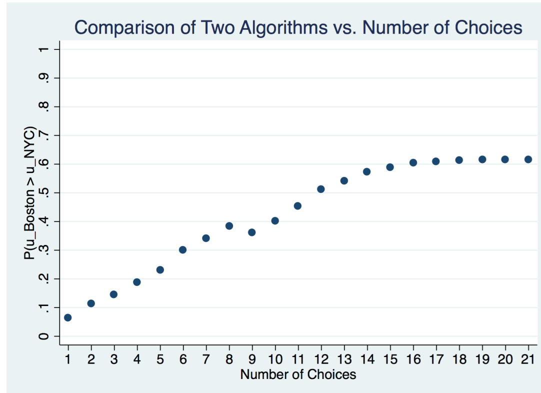


Figure 13: *Performance of Boston algorithm*



We again observe the same relationship between number of choices and average utility for both algorithms that we did in scenario one. We also expect that utilities are lower in this scenario. If students care mostly about academic quality of schools over distance then students' preferences will be more similar and there will be more competition for the same schools. The utilities at higher numbers of choices are lower, but the difference is small.

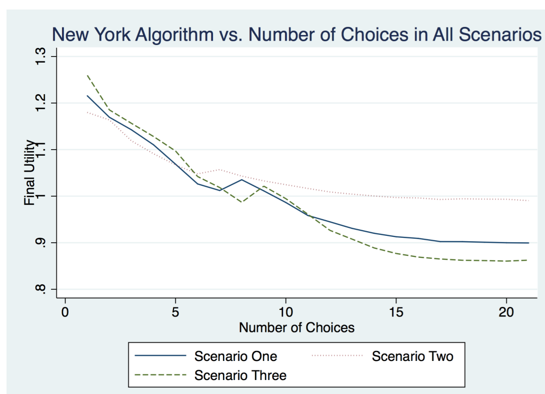
Figure 14: *Comparison of Boston and New York algorithms*



The Boston algorithm performs better than the NYC algorithm in this scenario when number of choices is larger than 12. This is higher than it was in scenario one (11) and two (7). The increase in probability is relatively constant when number of choices is small and becomes increasingly small when number of choices is closer to 21.

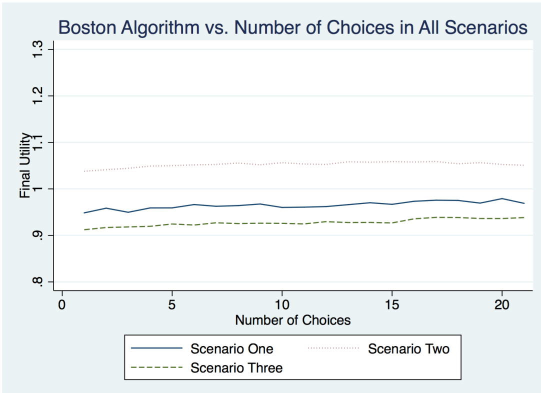
4.3 Comparison of Scenarios

Figure 15: *Performance of New York algorithm in each scenario*



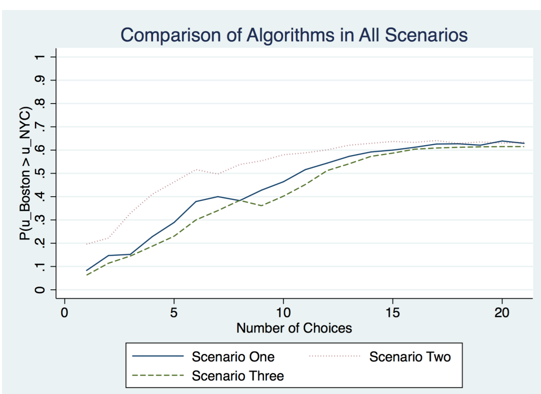
As mentioned earlier, we expected the utilities in scenario one to be smaller than those in scenario two and larger than those in scenario three. Interestingly, we see that the opposite is true for small number of choices although the difference between the scenarios is small. At the other end of the spectrum, we see the ordering of the scenarios that was expected and that the difference between them is larger.

Figure 16: *Performance of Boston algorithm in each scenario*



For the Boston algorithm the we also expected scenario one to do worse than scenario two and better than scenario three. In this case the results are as expected across all numbers of choices and the differences between them is about the same as in the New York algorithm when number of choices is close to 21.

Figure 17: *Comparison of Boston and New York algorithms in each scenario*



We see that the probability that a random utility taken from the Boston distribution is larger than one taken from the New York distribution increases as the number of choices increases in all three scenarios. When the number of choices is small the Boston algorithm performs best in scenario two and worst in scenario three. This difference between the three scenarios becomes much smaller when the number of choices is larger.

5 Future Work

The methodology with which we create the acceptance rates for the Boston algorithm requires additional study. In practice we normally see acceptance rates for schools remain largely the same from year to year with sometimes a small downward trend. However currently in the simulation the acceptance rates for schools can vary tremendously from year to year. The program in the future should be modified to find an equilibrium acceptance rate for each school that only varies a small amount over time.

It would also be beneficial to the general study of the algorithms to create additional districts based on other cities such as New York or Chicago to see how performance of the algorithms varies in these cities compared to Washington D.C. We would also like to study how characteristics of the school district such as density, demographics, average commute times and average school qualities affects how well the algorithms are able to match students to their schools of choice. Currently with just the data based on Washington we are only able to test out some of these effects by altering how students weigh their preference for academic quality vs. distance to schools.

Another thing to study would be to look at other ways in which students' priority numbers at schools are determined. In this paper for all simulations we had students' priorities at schools randomly determined as in a lottery. However, many school systems do not generate priorities completely randomly and instead also have schools creating preferenes for certain students. One example of this is allowing students to be guaranteed their "home school," which is generally the school they are located closest too. One interesting alteration to the algorithms would be to add in a check so that if a student lists their neighborhood school as their top choice, their admittance would be guaranteed.

We would also like to examine the performance of a third method of matching students to schools. This third method would be given by running a linear programming problem solver that is set up to maximize student utility. It would be interesting to see if it does beat, and if so by, how much, the New York and Boston algorithm distributions in terms of maximizing student utilities. We would also look at how many students are receiving their top choices. Running a linear program in real life may prove more difficult, as it would be harder to get students to express how much they like each school with scores as they do in this simulation. While the Boston and New York algorithms are run using only ordinal preferences, additional information is required in this case.

References

[1] Abdulkadirolu, Atila, Parag A. Pathak, and Alvin E. Roth, "The New York City High School Match, *American Economic Review*, Papers and Proceedings, 95,2, May, 2005, 364-367

[2] Abdulkadirolu, Atila, Parag A. Pathak, Alvin E. Roth, and Tayfun Snmez, "The Boston Public School Match, *American Economic Review*, Papers and Proceedings, 95,2, May, 2005, 368- 371

[3] Abdulkadirolu, Atila, Parag A. Pathak, and Alvin E. Roth, "Strategy-proofness versus Efficiency in Matching with Indifferences: Redesigning the NYC High School Match, *American Economic Review*, 99, 5 (December), 2009, 1954-1978.

[4] Hoxby, Caroline M. "School Choice and School Productivity (Or, Could School Choice be a Rising Tide that Lifts All Boats?," in C. Hoxby, ed. The Economics of School Choice, Chicago: *University of Chicago* Press, 2003

[5] Hoxby, Caroline M. "School Choice and School Competition: Evidence from the United States, *Swedish Economic Policy Review*, 10.2, 2004

[6] Roth, A.E. "The National Resident Matching Program as a labor market," JAMA. *Journal of the American Medical Association*, April 3, 1996, 275, No. 13, (Pulse), 1054-1056

APPENDIX

Table 1: *Performance of both algorithms after one round (scenario one)*

num_choice	u_1rd_nyc	u_1rd_bos
1	0.2517042	0.3145368
2	0.3626592	0.420395
3	0.4366921	0.452341
4	0.4969363	0.4427465
5	0.5736864	0.4402631
6	0.6575949	0.4682161
7	0.7327407	0.5080027
8	0.7622412	0.5672828
9	0.7886415	0.5525326
10	0.8223806	0.562108
11	0.8537822	0.5782483
12	0.8694939	0.5688906
13	0.8810693	0.5778078
14	0.8913876	0.5847954
15	0.8932809	0.5938176
16	0.8984443	0.5843971
17	0.9006987	0.5960944
18	0.8999076	0.6311955
19	0.8997027	0.6252746
20	0.9016429	0.6670133
21	0.9017325	0.6638835

Table 2: *Performance of both algorithms after all students matched (scenario one)*

num_choice	u_final_nyc	u_final_bos
1	1.215443	0.9484768
2	1.169423	0.9585332
3	1.142475	0.9497481
4	1.110334	0.9590523
5	1.06858	0.9591858
6	1.02627	0.9663354
7	1.012099	0.9626036
8	1.035211	0.9640842
9	1.011295	0.9674831
10	0.9863075	0.960128
11	0.9584109	0.9607467
12	0.944769	0.9619863
13	0.9308	0.966096
14	0.9202216	0.9702312
15	0.9127994	0.9667973
16	0.9092099	0.9733698
17	0.9023928	0.9756593
18	0.9022461	0.9751843
19	0.901062	0.9694586
20	0.8998791	0.979116
21	0.8994416	0.9689388

Table 3: *Probability that a random utility after one round is greater from the Boston algorithm distribution than the New York algorithm distribution (scenario one)*

# Of Choices	P(BOS > NYC)	#	P(-)	#	P(-)	#	P(-)
1	0.524	6	0.407	11	0.401	16	0.392
2	0.506	7	0.399	12	0.395	17	0.413
3	0.478	8	0.411	13	0.397	18	0.427
4	0.449	9	0.406	14	0.390	19	0.425
5	0.423	10	0.400	15	0.401	20	0.455
						21	0.458

Table 4: *Probability that a random final utility is greater from the Boston algorithm distribution than the New York algorithm distribution (scenario one)*

# Of Choices	P(BOS > NYC)	#	P(-)	#	P(-)	#	P(-)
1	0.083	6	0.379	11	0.516	16	0.612
2	0.147	7	0.400	12	0.544	17	0.626
3	0.152	8	0.383	13	0.573	18	0.627
4	0.228	9	0.428	14	0.592	19	0.621
5	0.289	10	0.464	15	0.600	20	0.639
						21	0.629

Table 5: *Performance of both algorithms after all students matched (scenario two)*

num_choice	u_final_nyc	u_final_bos
1	1.179759	1.037917
2	1.163682	1.04127
3	1.118756	1.044321
4	1.091207	1.049005
5	1.066951	1.049957
6	1.047922	1.051647
7	1.05699	1.052701
8	1.043078	1.055551
9	1.032773	1.051864
10	1.024574	1.056332
11	1.016506	1.053511
12	1.00896	1.052553
13	1.004308	1.058327
14	1.000359	1.057381
15	0.9970535	1.058653
16	0.9961119	1.057882
17	0.992845	1.058807
18	0.9943085	1.053988
19	0.9936461	1.056425
20	0.9933376	1.052554
21	0.9903981	1.050788

Table 6: *Probability that a random utility is greater from the Boston algorithm distribution than the New York algorithm distribution (scenario two)*

# Of Choices	P(BOS > NYC)	#	P(-)	#	P(-)	#	P(-)
1	0.196	6	0.516	11	0.588	16	0.633
2	0.222	7	0.497	12	0.601	17	0.641
3	0.329	8	0.538	13	0.621	18	0.629
4	0.410	9	0.554	14	0.629	19	0.635
5	0.463	10	0.580	15	0.637	20	0.630
						21	0.634

Table 7: *Performance of both algorithms after all students matched (scenario three)*

num_choice	u_final_nyc	u_final_bos
1	1.259419	0.9122034
2	1.185193	0.9169526
3	1.156377	0.9182253
4	1.128146	0.9195291
5	1.096959	0.9245012
6	1.042131	0.9222951
7	1.018513	0.9271491
8	0.9867831	0.9253723
9	1.021484	0.9262146
10	0.9945677	0.9258682
11	0.961086	0.9247387
12	0.9264925	0.9295763
13	0.9076727	0.9276634
14	0.8887671	0.92784
15	0.8768676	0.9268346
16	0.869137	0.9356254
17	0.8650405	0.9387759
18	0.8622697	0.9384607
19	0.8615701	0.9363257
20	0.8604784	0.9362492
21	0.86252	0.9383148

Table 8: *Probability that a random utility is greater from the Boston algorithm distribution than the New York algorithm distribution (scenario three)*

# Of Choices	P(BOS > NYC)	#	P(-)	#	P(-)	#	P(-)
1	0.063	6	0.300	11	0.452	16	0.604
2	0.114	7	0.340	12	0.512	17	0.609
3	0.145	8	0.384	13	0.541	18	0.612
4	0.187	9	0.361	14	0.573	19	0.614
5	0.230	10	0.402	15	0.587	20	0.615
						21	0.615