3219B014 Hayashi Yuji
The final report in the class of applied game theory

The comparison of total utility between proposers and reviewers achieved by the DA algorithm in a matchmaking party

Contents

1. Introduction

In matching problems composed of multiple proposers and reviewers having preferences over each other, the deferred acceptance algorithm (the DA algorithm) is well known for its positive aspects. The computation of the algorithm is not time-consuming, but the algorithm satisfies some desirable axioms we will explain later. This section explains how the algorithm works briefly and what axioms it satisfies.

1.1 The DA algorithm

The following explanation regarding the algorithm comes from Sakai (2018). In the beginning, let us define that for a proposer p, a reviewer r is acceptable if $r \succ_p p$, and for a reviewer r, a proposer p is acceptable if $p \succ_r r$. Then, for a proposer, "acceptable reviewers" means the reviewers who have not rejected him before and who are acceptable for him.

The DA algorithm works as follows. In the first step, each proposer proposes to a reviewer who is acceptable and at the top of his preference. If a proposer does not have any acceptable reviewers, he never proposes to any reviewers, meaning he gets alone. Each reviewer defers a proposer who is acceptable and at the top of her preference. She rejects other proposers. If a reviewer does not take proposals from proposers who are acceptable for her, she gets alone.

In the later steps $s \geq 2$, all the proposers who were rejected by reviewers at step $s-1$ propose to reviewers who are acceptable and at the top of their preferences at this step. If a proposer does not have any acceptable reviewers, he gets alone. Regarding the behaviors of reviewers, there are four patterns. First, if a reviewer kept deferring a proposer at step $s-1$ and does not take proposals from other proposers at this step, she keeps deferring the proposer. Second, if a reviewer did not keep deferring a proposer at step $s-1$ and does not take proposals from any proposers at this step, she gets alone. Third, if a reviewer kept deferring a proposer at step $s-1$ and takes proposals from n proposers at this step, she defers the proposer who is at the top of her preference in $n+1$ proposers. She rejects all the n proposers who are not deferred or acceptable at this step. Fourth, if a reviewer did not keep deferring a proposer at step $s-1$ and takes proposals from n proposers at this step, she defers the proposer who is at the top of her preference in n proposers. She rejects all the $n-1$ proposers who are not deferred or acceptable at this step.

The procedure is terminated once no proposer proposes to reviewers. Each proposer can propose to each reviewer only one time, meaning that the maximum number of iterating this algorithm is limited by the number of participants.

1.2 The axioms of the DA algorithm

The DA algorithm satisfies the axioms: individual rationality, stability, strategy proofness, and optimality for a proposer group. Individual rationality here means that no participant will be matched with an unacceptable person. If a participant prefers being alone to getting matched with someone, the participant does not have to be getting matched with that person. Then, stability here means no pair blocks the matching result obtained by the algorithm. There is no more profitable private matching between a certain proposer and a reviewer without joining the matchmaking process than the matching result obtained by the algorithm. This implies efficiency, too. Strategy proofness means each participant cannot improve his or her utility by telling a false preference. Each participant tells their true preferences. Finally, optimality for a proposer group means that the achieved matching result by the algorithm is the most profitable for a proposer group in all the possible stable matchings.

1.3 The motivation of this report

As discussed previously, the algorithm is known for obtaining the most optimal matching result for a proposer group. However, it seems natural that the reviewer's bargaining power would affect the optimality of the matching results for each group. And a reviewer's bargaining power would depend on the degree to refuse a proposal from a proposer. Therefore, this report investigates how the degree to which the proposers and the reviewers prefer being alone to getting matched with someone affects the difference in the total utility of both groups in a matchmaking party environment.

2. Experiments

2.1 Problem setting of the participants and their preferences

This section formulates the environment for a hypothetical matchmaking party. We define proposers and reviewers as a 10 participants tuple respectively:

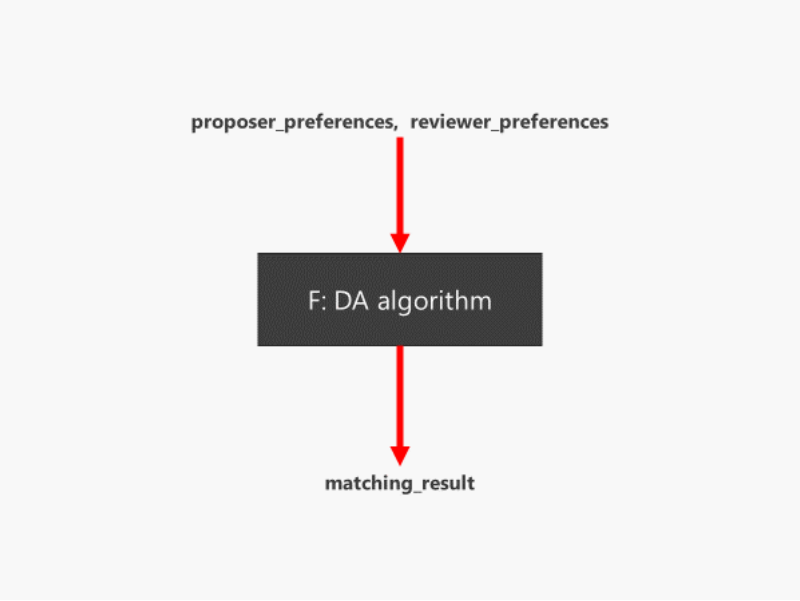
proposers = (1,2,3,4,5,6,7,8,9,10), *reviewers* = (11,12,13,14,15,16,17,18,19,20)

Then, we assign the preferences over the participants of the opponent group to each participant by generating the randomly ordered lists of *proposer_preferences* and *reviewer_preferences*. Finally, we insert their own numbers into their preferences to represent the order of preference for being alone. We change the order of preference for being alone from the top to the bottom of their preferences by each experiment and compare the total utility of both groups. Besides, in a matchmaking situation, we assume that all the participants tell their true preferences, behaving non-strategically.

2.2 Our framework

Figure 1 illustrates the overview of our framework.

Figure 1: The overview of our framework



We can view the DA algorithm as a function that maps the dictionaries of the lists *proposer_preferences* and *reviewer_preferences* to the dictionary of *matching_result*.

Table 1 shows an example of *proposer_preferences*, *reviewer_preferences,* and *matching_result* when the participants prefer being alone at the 4th of their preferences.

Table 1: An example of participants' preferences and a matching result when they prefer "being alone" at the 4th of their preferences

proposer_preferences	{1: [17, 18, 20, 1, 11, 14, 15, 19, 12, 16, 13], 2: [19, 16, 18, 2, 20, 17, 11, 14, 13, 15, 12], 3: [20, 18, 14, 3, 17, 12, 19, 16, 13, 11, 15], 4: [17, 20, 12, 4, 19, 13, 15, 14, 18, 16, 11], 5: [14, 19, 16, 5, 20, 11, 17, 13, 18, 12, 15], 6: [19, 13, 16, 6, 14, 18, 20, 15, 11, 17, 12], 7: [19, 18, 17, 7, 16, 15, 20, 12, 13, 11, 14], 8: [15, 18, 20, 8, 14, 19, 17, 11, 16, 13, 12], 9: [20, 15, 18, 9, 11, 12, 13, 19, 16, 17, 14], 10: [12, 11, 13, 10, 18, 19, 14, 17, 15, 20, 16]}
reviewer_preferences	{11: [10, 4, 5, 11, 9, 1, 3, 2, 6, 7, 8], 12: [1, 7, 10, 12, 3, 4, 6, 5, 8, 9, 2], 13: [5, 9, 2, 13, 4, 3, 8, 10, 1, 7, 6], 14: [5, 9, 10, 14, 1, 8, 3, 4, 7, 2, 6], 15: [2, 6, 3, 15, 10, 8, 9, 1, 5, 7, 4], 16: [1, 2, 4, 16, 9, 8, 10, 5, 3, 7, 6], 17: [8, 10, 9, 17, 6, 2, 4, 3, 1, 7, 5], 18: [8, 6, 7, 18, 10, 9, 3, 4, 5, 1, 2], 19: [4, 8, 6, 19, 3, 2, 7, 9, 5, 1, 10], 20: [8, 2, 5, 20, 9, 6, 4, 1, 3, 10, 7]}
matching_result	{1: 1, 2: 16, 3: 3, 4: 4, 5: 14, 6: 19, 7: 7, 8: 18, 9: 9, 10: 12, 11: 11, 12: 10, 13: 13, 14: 5, 15: 15, 16: 2, 17: 17, 18: 8, 19: 6, 20: 20}

The lists of *proposer_preferences* and *reviewer_preferences* were generated randomly in each experiment. The pairs of the same numbers mean that the participants get alone, instead of getting matched with someone.

2.3 Evaluation metrics

To evaluate the satisfaction of each participant with a matchmaking party, we define the utility of each participant given *proposer_preferences*, *reviewer_preferences*, and *matching_result*. Here, we calculate the utility as:

11 - the order of a matched partner from the left in his/her preference

Therefore, the maximum utility of each participant can be 10 and that minimum can be 0.

For example, in the example of Table 1, Proposer 1 gets alone. His order of preference for being alone is at the 4th, meaning the index of being alone in his preferences is 4. Therefore, the utility of Proposer 1 given the fulfilled result and his preferences is:

11-4=7

In the same way, Proposer 2 gets matched with Reviewer 16, and Reviewer 16 is at the 2nd in his preferences. Therefore, the utility of Proposer 2 is:

11-2=9

Finally, we sum up their utilities and obtain the total utilities of both groups in each experiment. Thus, the maximum total utility of each group can be 100 and that minimum can be 0. We iterate this experiment 1000 times for each different order of preference for being alone in the participants' preferences and obtain the results like Figure 2.

Figure 2: The results table when the participants prefer being alone at the 4th of their preferences

pref_alone	experiment_id	total_utils_proposers	total_utils_reviewers
4	1	83	80
4	2	83	81
4	3	81	80
4	4	81	81
4	5	84	83
...	...	...	...
4	996	81	79
4	997	81	83
4	998	83	84
4	999	79	80
4	1000	86	87

Figure 2 shows the results table when the participants prefer being alone at the 4th of their preferences.

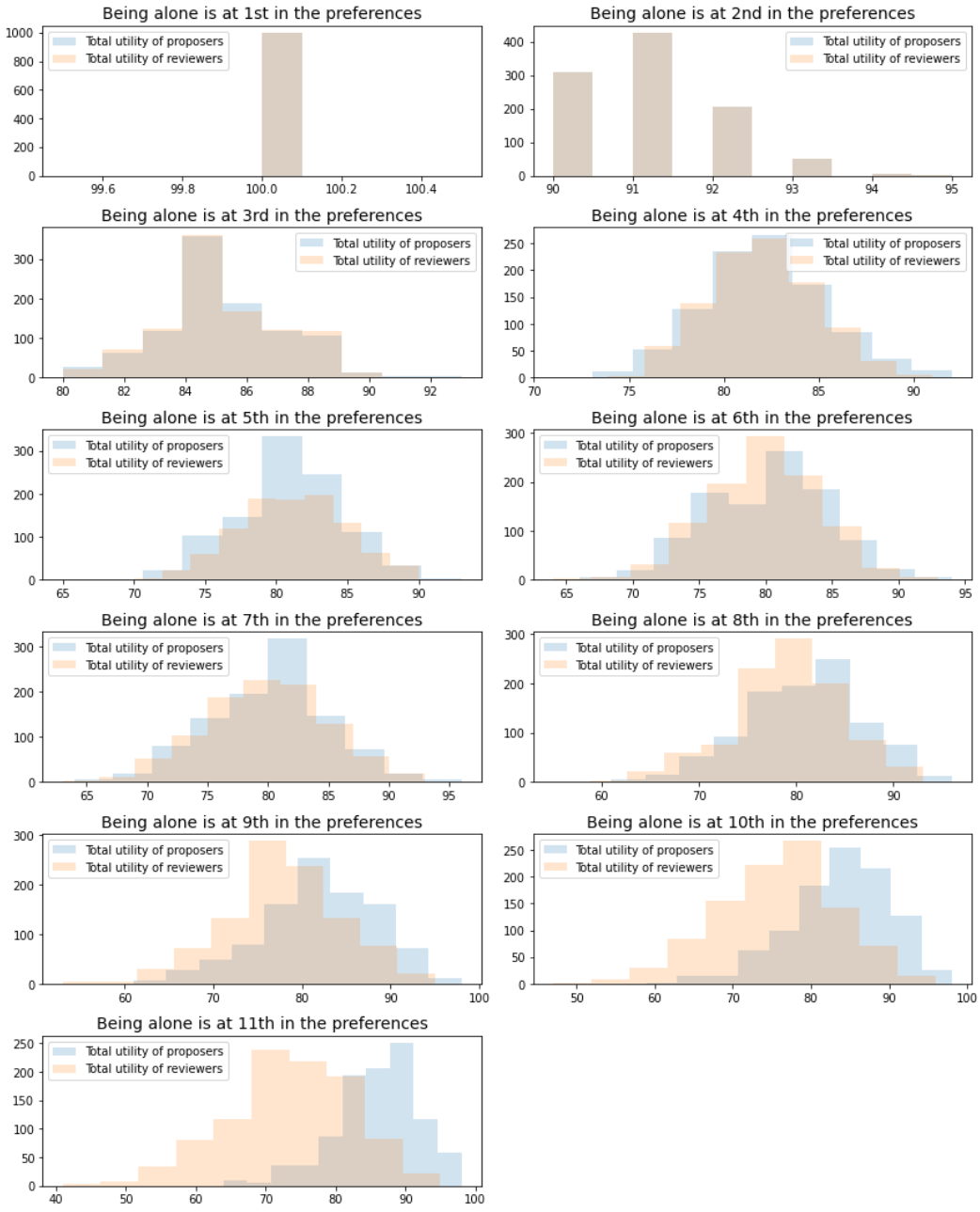
Next, we analyze the distributions of the total utilities of both groups by visualization and statistical testing. Visualization tells us how different the total utility is in the proposers and the reviewers. Thus, we check the difference here is statistically significant. For this purpose, we conduct the Wilcoxon Signed-Rank test. The Wilcoxon Signed-Rank test has its null hypothesis that the paired samples from the two series follow the same distribution, meaning the difference of two samples d = s1 – s2 is distributed around zero symmetrically (Scipy.org, 2021). We also set up the alternative hypothesis that the difference is greater than 0.

3. Results

3.1 Visualization of the total utility distributions

Figure 3 visualizes the total utility distributions generated from 1000 experiments for each different order of preference for being alone in the participants' preferences.

Figure 3: Histogram: The total utility of the two groups



When the order of preference for being alone is at the 1st or 2nd in the preferences, the total utilities of both groups seem to follow the same distribution. These two cases have almost no

matching pair because the participants prefer being alone to getting matched with almost all the opponents.

When we see the total utility distributions, the visualization suggests that the higher the order of preference for being alone, the smaller the difference between the total utility of the proposers and the reviewers. This result seems inconsistent with the proposers' optimality in the algorithm. The DA algorithm should yield a matching result that is the most optimal for the proposer group in all the possible stable matchings. If so, the difference between the total utilities of both groups must not be zero. When the order of preference for being alone is lower, the participants want to get matched with anyone rather than being alone. In these situations, proposers can still propose to reviewers they truly prefer from the top of their preferences, while reviewers need to defer new proposers if they have not taken proposals from other proposers whom they prefer to previously. Therefore, proposers have the advantage to maximize their total utility. However, our results imply that if reviewers prefer being alone to getting matched with enough proposers, this seems to offset that advantage.

3.2 Statistical testing on the total utility distributions

Table 2 shows the result of the Wilcoxon Signed-Rank test on the distributions of the total utility of the two groups. The difference is calculated as the total utility of the proposers – the total utility of the reviewers.

Table2: The result of the Wilcoxon Signed-Rank test on the distributions of the total utility of the two groups (proposers - reviewers)

The order of preference for being alone	Wilcoxon statistics	P-value
1	-	-
2	-	-
3	118699.5	0.300
4	176084.0	0.297
5	205615.5	0.377
6	224602.5	0.174
7	237104.0	0.004
8	271829.5	0.000
9	332755.5	0.000
10	394861.0	0.000
11	442964.5	0.000

From the P-values of Table 2, we cannot reject the null hypothesis that the total utility of the proposers is the same as that of the reviewers in the 0.1 significance level when the order of preference for being alone is higher than the 6^{th} in the participants' preferences. Thus, we can say the total utilities of both groups are not different when the proposers and the reviewers prefer being alone to getting matched with enough opponents. When the order of preference for being alone is the 1^{st} and 2^{nd} in the preferences, we cannot get the results of the test because the distributions of both groups are totally identical.

4. Conclusion

From the experiments, we found that the total utilities of proposers and reviewers are not different when the order of preference for being alone is equal to or higher than the 6th in a matchmaking party that has ten proposers and ten reviewers on the DA algorithm. Thus, in our simulations, the optimality for the proposer group on the DA algorithm seems to be offset by the refusal power of the reviewers.

Besides, it is not unnatural that we assume the order of preference for being alone is between the 3rd to the 6th in the participants' preferences in actual matchmaking parties. Therefore, if we ask the participants to write down their preferences over all the opponents including the option of not getting matched with anyone, and employ the DA algorithm, the total utilities of proposers and reviewers do not seem so different in a practical matchmaking situation.

Our limitations are not showing the theoretical support for our experimental results. By referring to theoretical articles, our experimental results should be explained more. The proceeding works are required to fill this gap.

5. Bibliography

T, Sakai. (2019). *Introduction to Market Design*. Minerva Shobo.
Burcă, C. (n.d.). Deferred Acceptance Algorithm. Retrieved July 20, 2021, from https://gist.github.com/scribu/104ec4ba54207db8c6e8
Scipy.org. (2021, 4 26). scipy.stats.wilcoxon. Retrieved 05 25, 2021, from scipy.stats.wilcoxon - SciPy v1.6.3 Reference Guide: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.wilcoxon.html
Wilcoxon, F. (1945). Individual Comparisons by Ranking Methods. Biometrics Bulletin, 1(6), 80-83.