Fair Housing Assignment

EE382V Social Computing Fall 2018

Beth Richardson (blr2245), Tiffany Tillett (tat573), Porter Perry (perrypl1), and Javier Palomares(jp46396)

Abstract

The Housing Assignment problem deals with assigning a set of agents to a set of houses: each agent being assigned exactly one house. The algorithm for housing has a goal to assign an agent to the most desirable house based on preference. The base algorithm leads to a different assignment based on the initial ownership of each house; the fair housing assignment presented in this paper attempts to provide a fair housing assignment in which all owners, regardless of initial ownership have an equal chance to own their preferred house.

Overview of the Housing Assignment Algorithm

A large proportion of the lecture material for this class has been either directly or indirectly related to subject of matching, it's theories, and various algorithm solutions. Matching problems can involve two-sided matching, in which both sides are agents, such as men and women or employers and laborers, are to be matched with each other. Matching can also be one-sided, in which there is one side of agents, and another side of objects (or houses), that are to be assigned (matched or allocated) to them. The later is the subject of this term project.

For the housing allocation problem, we have N houses and N agents and we want to assign each agent to a single house. Each agent has a strict preference list for each of the N houses. The resultant allocation is considered to be "core" if there is no subset of agents that could all get a more preferred house by swapping with each other.

As discussed in class, David Gale's Top Trading Cycle Algorithm (TTCA) addresses the original housing allocation problem in which each agent starts out with a house. The TTCA will find the unique core allocation in which each agent gets their a house no worse than the house that they started with on their preference list. What about the case where agents do not start out with a house? How many core allocations for this new problem exist? If multiple core allocation exists, how can they be compared? Can one allocation be determined to be more fair than another? Exploring the answers to the above questions lead us to the algorithms, experiments, and results in this report.

Bias in the Original Algorithm

The original housing allocation algorithm does not try to ensure that all agents are given a reasonably fair assignment in relation to the other agents. Instead, it leverages the initial ownership of the houses and simply tries to find an assignment that does not lead to any agents wanting to swap houses. This can lead to some agents having an unfair advantage.

The original housing algorithm can lead to those agents with the most highly rated houses getting priority in the selection of their preferred house. In the below example, the algorithm does not change anything and each agent will end up with their original house. The algorithm works by looking for and eliminating cycles where each agent initially points to its most preferred house and each house initially points to its current owner.

```
4
1 2 3 4
2 4 3 1
1 3 4 2
2 3 1 4
```

Figure 1: Example of initial assignment and preference that leads to bias (found in test3.txt)

We see a cycle between Agent 1 and House 1 and between Agent 2 and House 2, so these assignments are made. Removing both Agent 1 and Agent 2 as well as House 1 and House 2, we see a cycle between Agent 3 and House 3. This leaves Agent 4 with House 4 even though it is Agent 4's last choice. An assignment where two agents get their first choice, one agent gets their second choice and the final agent gets their last choice is hard to label as fair. However, this is the result that we end up with because of the priority that we give to an agent who currently owns a desirable house.

The Fair Assignment Algorithm

Our fair assignment algorithm hopes to remove this bias by providing a means to provide each agent with the opportunity to procure their highest priority house without using the agents' initial house ownership to resolve conflicts. We also provide the means to compare assignments based on a fairness score and use this score to iterate through potential assignments and select the one that provides the fairest result for all agents.

Initialization

We do not provide any initial ownership for each agent in our algorithm. This ensures that no agent is given a preference based on a previous ownership in a house and instead provides for all agents to have an equally fair final house assignment.

Breaking Ties

In each matching strategy, there must be an implementation for how to break ties between groups of agents who want the same house. In our fairness assignment algorithm, we break ties using the index of the agents in the priority ordering to determine who wins. We feel that this technique, which is not based on previous ownership, is a fair strategy for breaking ties without preferring one agent over another given that we try all possible priority orderings.

Generating and Comparing Assignments

Within any given priority ordering, the agents that have a lower index are going to be preferred over the other agents when breaking ties. Depending on the ranking of houses and the agent's preferences, the strategy can lead to certain agents being much preferred over others given a particular priority ordering. Because we prefer an assignment algorithm that does not provide preference to any given set of agents, our algorithm provides the means to generate all possible priority orderings, solve for each of them and then compare all solutions to select the assignment that results in the fairest allocation. In order to accomplish this, we use a measurement that we created called the fairness score.

Fairness Score

In our assignment algorithm, we iterate through all possible assignments and rank them using a fairness score. This score is calculated by iterating through all agents and summing the absolute value of the difference of the rank of the house they were assigned to the average rank assigned to all agents. We consider a potential assignment with a lower fairness score to be a fairer assignment. This basically attempts to minimize the deviation from the average rank that each agent is assigned. In an optimum solution, each agent is assigned the same rank house, which would be the average rank and then the fairness score would be 0. For example, if an agent is assigned a house that is their 5th rank house, but the average rank for all other agents is 3, that agent would add 2 to the fairness score for that assignment. An equation defining fairness is given in Figure 3.

```
public double getFairnessScore()
{
  double fairness = 0.0;
  double avgRank = getAvgRank();
  Set<Agent> agents = agent2HouseMap.keySet();
  for(Agent agent: agents)
  {
     List<House> preferences = agent.preferences;
     House assignedHouse = agent2HouseMap.get(agent);

     int preferenceIndex = getPreferenceIndex(assignedHouse,preferences);
     fairness += Math.abs(preferenceIndex - avgRank);
  }
  return fairness;
}
```

Figure 2: Calculating the fairness score for a housing allocation

$$fairness = \sum_i | rank_i(H) - average(rank_j(H)) |$$

Figure 3: Fairness is defined as sum over all agents of the absolute value of the difference between an agent's assigned house rank and the average assigned house rank.

Initialization and Matching Strategies

In order to arrive at the previously mentioned algorithm, we modeled the housing assignment problem with a set of Initialization and Matching strategies. A given assignment algorithm provides a base initialization strategy to set initial agent ordering and house ownership. The assignment algorithm will also have a matching strategy to determine how to generate the final matching. The fair assignment is made up of the No Owners Initialization strategy and the Default Matching strategy, with the addition of a step to generate all possible agent orderings and a mechanism for comparing the final assignments using the fairness score.

We included several other experimental initialization and matching strategies in our source code for additional review. These strategies will be detailed below.

Initialization Strategies

No Owners Initialization

This is the strategy we selected for our fair housing assignment algorithm because it does not provide any initial preference toward any agent based on existing ownership. In this strategy, all houses are presented without initial owners and agents are assigned without concern for this initial ownership.

Owners Initialization

This initialization strategy merely assigns each agent to the house with the same index as the agent. This strategy aligns with the initial housing allocation algorithm. However, this strategy was not selected for our algorithm as it suffers from the preference of some agents over others.

Middle Initialization

This initialization strategy attempts to minimize the impact of initial ownership by assigning all agents to their middle ranked housing preference to start. This seemed to be a fair strategy for initial assignment of home; however, we did not use this in any of our final implementations because it did not add much benefit over the owners initialization and still provided a bias based on initial ownership.

Matching Strategies

Default Matching

This is the matching strategy we selected for our fair housing assignment algorithm. In it, we give each agent their first choice and then break ties using the index of the agent in the agent priority provided. In our fairness algorithm, we set these priorities using an arbitrary ordering of

the agents. We then iterate through all possible orderings and rank the final solutions based on fairness score to determine which solution is most fair for all agents.

Poorest Agent Wins Matching

In this matching strategy, ties are broken by giving preference to the agent who has a house assigned that is the least desirable. In this strategy, the agent that will gain the most by the swap is preferred. However, this strategy prefers those with a lower initial housing assignment rather than optimizing for overall fairness.

Fair Owner Matching

This strategy is very similar to the poorest agent wins matching strategy. However, if there is an agent that already owns the desired house, that agent will get the house. Otherwise, it follows the same strategy as the Poorest Agent Wins Matching.

Results

We did most of our experiments using a simple four agent example.

```
4
3 2 1 4
1 2 3 4
4 2 1 3
1 3 2 4
```

Figure 4: Four Agent example (found in test1.txt)

In this example, we can clearly see that not all agents can get their first preference since both Agent 2 and Agent 4 prefer House 1. If we break the tie by giving the house to Agent 4, Agent 2 will move on to House 2 where there is no conflict, so we have our allocation: Agent 1, House 3; Agent 2, House 2; Agent 3, House 4; Agent 4, House 1. If we instead break the tie by giving the house to Agent 2, Agent 4 will move on to House 3 creating another conflict with Agent 1. Again, we have two options that we can explore. By trying all possible agent orderings, we end up with the below three possible allocations where the number in parentheses represents the rank of the assigned house for that agent.

1. A1, H3 (1); A2, H2 (2); A3, H4 (1); A4, H1 (1)
2. A1, H3 (1); A2, H1 (1); A3, H4 (1); A4, H2 (3)
3. A1, H2 (2); A2, H1 (1); A3, H4 (1); A4, H3 (2)

We can then calculate the fairness scores for each of those allocations to determine which is best. Since the average rank for the first allocation is 1.25, we sum 0.25 each from Agents 1, 3 and 4 with 0.75 from Agent 2 to get a fairness score of 1.5. Since the average rank from the

second allocation is 1.5, we sum 0.5 each from Agents 1, 2 and 3 with 1.5 from Agent 4 to get a fairness score of 3. Since the average rank from the third allocation is also 1.5, we sum 0.5 from all agents to get a fairness score of 2. We can see that the first allocation has the lowest fairness score, so that is the allocation that we go with.

```
House 4 is owned by agent 3
House 1 is owned by agent 4
House 2 is owned by agent 2
House 3 is owned by agent 1
Average rank of allocated Houses to Agents is 1.25
```

Figure 5: Output of running Fair Housing Assignment on the example in Figure 4

We can refer back to the initial example provided in Figure 1 to illustrate how our algorithm is more fair. The original algorithm will simply assign each Agent to their original houses. However, this means that two agents get their first choices, one gets their 2nd choice and one gets their last choice. This provides an average rank of 2 and fairness score of 4. In contrast, we can run our algorithm on this same example and get:

```
House 4 is owned by agent 2
House 1 is owned by agent 3
House 2 is owned by agent 1
House 3 is owned by agent 4
Average rank of allocated Houses to Agents is 1.75
```

Figure 6: Output of running Fair Housing Assignment on the example in Figure 1

Here, one agent gets their first choice and the other three agents get their second choices. This provides us with a fairness score of 1.5. This is both more socially optimal and more fair than the original algorithm.

With additional time, we could explore the tradeoffs between fairness and optimality. We could also experiment more with our calculation of fairness score. For example, there is another valid allocation that gives two of the agents their first choices and the other two their second choices. This provides an average rank of 1.5 which is more socially optimal than the result given, but not as fair.

Related Work

In the paper titled “Random Serial Dictatorship and the Core from Random Endowments in Housing Allocation Problems”, Abdulkadiroğlu and Sönmez examine matching mechanisms for both matching allocation and matching market problems, the one difference being whether each agent starts out with a house (referred to as an endowment) or not. They examine the resulting

solutions from solving both problems with a random uniform distribution, and show that interestingly both solutions are statistically equivalent.

Related to our fairness score, Irving et al. in the paper “Rank-Maximal Ranking” describes two algorithms for finding rank-maximal matching. A rank-maximal matching is defined as one in which the maximum possible number of applicants are matched to their first choice post, and subject to that condition, the maximum possible number are matched to their second choice post, and so on. The idea of rank-maximal matching is credited to be first introduced in the paper “Greedy Matching” by Irving.

References

Abdulkadiroglu, Atila, and Tayfun Sonmez. “Random Serial Dictatorship and the Core from Random Endowments in House Allocation Problems.” *Econometrica*, vol. 66, no. 3, May 1998, p. 689., doi:10.2307/2998580.

R.W. Irving, T. Kavitha, K. Mehlhorn, D. Michail, and K. Paluch. “Rank-Maximal Matchings.” Proceedings of SODA ’04, pages 68–75. ACM-SIAM, 2004.

Levin, Johnathan. “Lecture 4: House Allocation and Kidney Exchange.” Econ 136. web.stanford.edu/~jdlevin/Econ%20136/Lecture%204%20House%20Allocation%20and%20Kidney%20Exchange.pptx.

“Top Trading Cycle.” *Wikipedia*, Wikimedia Foundation, 22 Feb. 2018, en.wikipedia.org/wiki/Top_trading_cycle.