Understanding social welfare loss in homogeneous vs. heterogeneous dynamic matching markets

Mackenzie Kong-Sivert
mkongsiv@umd.edu

Ishan Sen
isen@umd.edu

Abstract

In this report, we implement a simple model of dynamic matching where planners have full criticality information and test them in homogeneous vs. heterogeneous dynamic markets. We extend the simple greedy model to one that may alternate strategies randomly at each time step and find that this strategy has a negligible difference on the overall loss. We analyze the results and propose extensions to better explore the field in more realistic settings.

1 Introduction

In this report, we extend upon studies of matching in a dynamic market and potential costs to social welfare (matching algorithm loss) that are a result of matching policies that are implemented. Dynamic matching markets such as ridesharing services (e.g. Uber, Lyft), organ exchange systems (e.g. kidney exchange), etc. typically seek to maximize their own utility which may come at a cost of social welfare. Literature shows how overall losses in social welfare are a result of non-optimal matching policies that lead to market fragmentation Akbarpour et al. (2020). As baseline cases for matching policies, we look at Greedy and Patient matching algorithms, discussed in detail in section 4. We will compare empirical results of the loss of our proposed matching algorithms to Greedy and Patient models via. simulations, and verify bounds on the loss of these algorithms based on theoretical findings from existing literature.

One major assumption that Akbarpour et al. (2020) make is that the market is homogeneous, i.e. the agents all are equally easy or hard to match. In some real-world settings this is not the case, as can be seen in kidney-exchange settings where type-O patients and non-O donors compete to match with type-O donors in the pool Roth et al. (2007). These compatibility restrictions may alter the structure of the market, and we would like to investigate how the basic greedy and patient matching algorithms perform when there is a divide in easy-to-match and hard-to-match agents in the market. To this end, we run preliminary experiments that compare the loss of Greedy and Patient matching algorithms in two settings with different proportions of easy-to-match and hard-to-match agents.

Furthermore, we implement a "Randomized" Greedy strategy where a planner matches greedily, but has full access to criticality information and with some small probability may decide to also match critical agents at every time-step. We find however, that this has a minimal effect on decreasing the overall loss. We propose extensions and alternatives to the discussed matching algorithms that may yield a lower overall loss.

2 Related Works

2.1 Dynamic Matching Markets

We heavily base our implementation and assumptions off of Akbarpour et al. (2020). The market assumption and setup as well as the simple baseline matching strategies implemented are based on their paper, and are explained throughout this report. A notable difference from Akbarpour's market assumption and other some markets is that agents are assumed to all be similar with regard to their compatibility. Indeed, other markets use this assumption for their models such as in real-time Ride Sharing Özkan & Ward (2020), however a more realistic scenario is where markets have agents that have varying levels of compatibility, a.k.a. heterogeneous dynamic markets. Real-world example of where such a heterogeneous markets can be found is in kidney-exchange, where certain patient-donor types are harder to match than others (Roth et al. (2007)).

2.2 Dynamic Matching Algorithms

There are a variety of matching algorithms that have been used and tested. While Akbarpour et al. implemented very simple Greedy and Patient models (that are also used in this report), others have investigated more sophisticated matching algorithms. For instance, Ashlagi et al. (2017) investigate chain matching, where the market aims to form a "chain" of agents who all satisfy each others' needs. In contrast to the simple algorithms used by Akbarpour et al. (2020) that find a random agent from the neighbours to match with, chain matching aims to match a group of agents at a time as opposed to only two at once. We acknowledge that there are many other variants of the patient and greedy matching algorithms, such as the Patient(α) Akbarpour et al. (2014), and these aim to improve upon the existing baseline models.

3 Market

This section describes the market model and assumptions that we use in our experiments, which is based off of Akbarpour et al. (2020). We assume that agents arrive into the market following a Poisson rate of m, i.e. within a time interval $[t, t+1]$ we expect m agents to enter the market with $m \geq 1$, and we denote the set of agents who actually enter the market at time t as A_t^n, where $|A_t^n| \leq 1$. Each agent in the market becomes critical following a Poisson rate of λ, normalized to 1. The set of critical agents is denoted by A_t^c. A critical agent leaves the market at time $t_0 + X$ if it is not matched by then, where X is an exponential random variable with a mean of 1. At a time step t, the pool of the market (set of agents) is defined as A_t and the pool size as $Z_t := |A_t|$. At time step t, an agent $a \in A_t$ leaves the market either by matching with another agent $b \in A_t$, irrespective if a is critical or not, and if a is critical and is not matched with another agent it leaves the market unmatched and *perishes*. An agent's length of stay in the market is denoted as its *sojourn*, defined as $s(a) := t'_a - t_a$, where t'_a is when agent a leaves the market and t_a is when that agent enters the market.

For an agent $a \in A_t$ to be matched, there needs to be another agent $b \in A_t$ that is compatible with a. Any two agents are independently compatible with probability d/m, where d scales the expected degree of each agent and m is the number of agents in the market. A compatible pair of agents are then matched by the planner that observes them (they are matched following a matching policy). The overall aim of the planner is to minimize the proportion of unmatched agents. This proportion can also be seen as the **loss** of any matching algorithm.

To further elaborate on the compatibility network created in the market, we have $E_t \subseteq A_t \times A_t$ be the set of compatible pairs of agents in the market at time t. Using this, the network at time t can be then be written as $G_t = (A_t, E_t)$. It is also assumed that compatible pairs persist over time, i.e. if two agents are compatible at a certain time, then they must be compatible at a latter time so long as they are both in the market. For each agent a, the neighbours of a can be written as $N_t(a) \subseteq A_t$.

Any matching algorithm identifies a set of edges $M_t \subseteq E_t$ that is a matching at time t, and the endpoints of the edges (if they exist) then leave the market. The following two matching algorithms that are discussed are different in when they decide to match an agent. Since looking at the global view, or the entire network, is computationally intractable, both matching algorithms focus on a more restricted local view by observing the immediate neighbours of each agent.

4 Models

The two baseline models we study are the Greedy and Patient algorithms, for which there are lower and upper bounds on the loss as derived by Akbarpour et al. (2020) given our market and agent definitions. In this section we summarize the strategies used by each model and the existing bounds computed for each. For all cases, we define the loss of any matching algorithm ALG as the proportion of unmatched agents to the total number of agents in the market. This loss can be seen defined as **L**(ALG) in the following, where $ALG(T)$ is the number of matched agents at time T:

$$\mathbf{L}(\text{ALG}) = \frac{\mathbb{E}[|A - ALG(T) - A_T|]}{mT}$$

The following subsections discuss bounds on the Greedy and Patient matching algorithms. The proofs for these bounds can be found in Akbarpour et al. (2020).

4.1 Greedy Matching Algorithm

The greedy algorithm attempts to identify a matched pair for an agent as soon as they enter the market. Note that this matching algorithm does not need information about when an agent becomes critical. For $d \geq 2$ and as $T, m \to \infty$, the loss **L**(Greedy) can be found to be bounded by:

$$\frac{1}{2d+1} \leq \mathbf{L}(\text{Greedy}) \leq \frac{\log(2)}{d}.$$

4.2 Patient Matching Algorithm

By contrast, a patient algorithm waits to attempt to match an agent until it becomes critical. The loss for this algorithm is also proven to have both an upper and lower bound in Akbarpour et al. (2020). For $d \geq 2$ and as $T, m \to \infty$:

$$\frac{e^{-d}}{d+1} \leq \mathbf{L}(\text{Patient}) \leq \frac{e^{-d/2}}{2}.$$

In the patient setting, we assume that the planner asks each agent if they are critical at time t, and we also assume that the agent reports truthfully all the time. We understand that this is not a fully realistic setting as the criticality information is not always accessible, and when it is it may not be truthful.

5 Our Model

For this project, we wrote a computational model of a dynamic matching market based on the framework presented in Akbarpour et al. (2014). Our aim was to determine the effect of changing several of the parameters of the model on the utility derived from switching strategies in a market, i.e., switching between a greedy algorithm and a patient one. The code for our model has versions written in both C++ and Python and is available at `https://github.com/mkongsivert/dynamic-matching`. We have uploaded the full Doxygen-generated documentation to `https://mkongsivert.github.io/dynamic-matching`, but we give an overview here.

Based on Akbarpour et al. (2014), we define a market to be a collection of agents and an Erdős-Rényi random graph representing which agents can be matched with each other (Erdős et al. (1960)). That is, it is a compatibility graph such that any two agents are compatible with each other with probability d/m. A market can either be greedy or patient, as defined in Section 4, and it can switch between strategies at any time step. An agent is an element in a market, and the market derives utility from matching agents before they become critical. Each agent has a lifespan, and once it has been in the market for a number of time steps equal to its lifespan, it becomes critical and will leave the market unmatched if it is not matched immediately.

The parameters which distinguish markets from each other are T, m, λ, d, δ, and, of course, whether its strategy is greedy or patient. T determines the number of discrete time steps for which the market

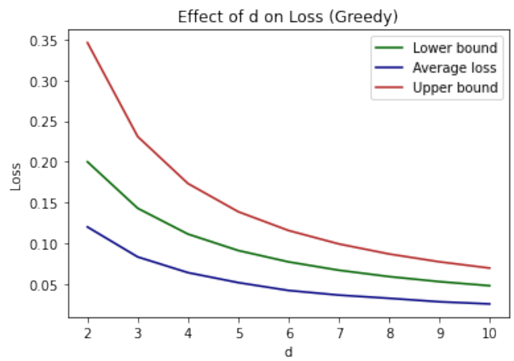

Figure 1: Loss of greedy matching algorithm with varying levels of d, $m = 200$, $T = 3000$.

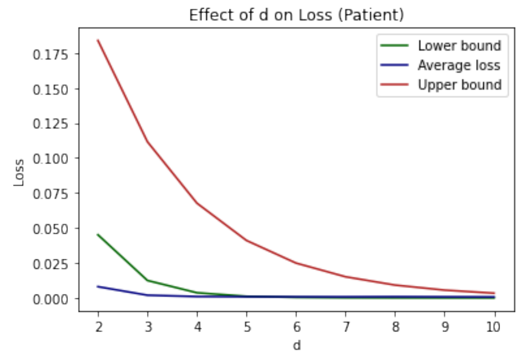

Figure 2: Loss of patient matching algorithm with varying levels of d, $m = 200$, $T = 3000$.

is open. Time steps are denoted $t_0, \ldots, t_n$, where $n = |T|$. At each time step, the market first eliminates all critical agents that cannot be matched. Note that at time t_0, a market has no agents.

After this, for each time step t_i, n_i agents enter the market, where n_i is determined by a Poisson distribution centered at m. Each agent is also assigned a lifespan, as mentioned above, and this is determined by a Poisson distribution centered at λ. Once an agent enters the market, we incorporate it into the Erdős-Rényi random graph by iterating through each of the other elements and determining whether they form a compatible pair together. Finally, the market iterates through the set of new agents or critical agents (depending on if the market is greedy or patient, respectively) and matches any compatible pairs.

In summary, T determines the length of time during which the market is open; m determines the number of agents that enter a market at each time step; λ determines the lifespan of each agent; d determines the likelihood that any two agents are compatible; and δ determines how much utility is lost while the agent waits to be matched.

6 Results

In this section we run preliminary tests using the baseline Patient and Greedy matching algorithms, as well as using some modifications on the nature of the agents in the market and also using a "Randomized" Greedy strategy, as defined later in this section.

6.1 Baseline Tests

We did not have very much time to run extensive tests, but we have run a few preliminary tests in order to determine the efficacy of our model. In particular, we ran two tests: one for the greedy algorithm and one for the patient algorithm. In both of them, we set $T = 2000$, $\lambda = 1$, and $m = 10$. Since Akbarpour et al. (2014) assumes that $\delta = 0$, we set $\delta = 0$ as well. Then, we ran 50 trials each on values of d ranging from 2 to 10 and averaged the total loss over all trials. We then plotted them against the theoretical upper and lower bounds for the loss given the respective algorithm. The results are pictured in Figures 1 and 2.

The reason for which our average loss values are lower than the theoretical lower bound can most likely be attributed to our low values for m and T, as we expect our actual loss to be within the bounds as $m, T \to \infty$. It is also important to note that our Patient algorithm used in these baseline performances assume that agents report truthfully when they become critical, which is not necessarily a realistic assumption to make at all times.

6.2 Heterogeneous vs. Homogeneous Markets

We run preliminary tests to compare the simple matching algorithms when initial assumptions of homogeneity to not hold. Here, we set a parameter p that determines whether an agent is easy-to-match with probability p or hard-to-match with probability $1 - p$. We define hard-to-match agents

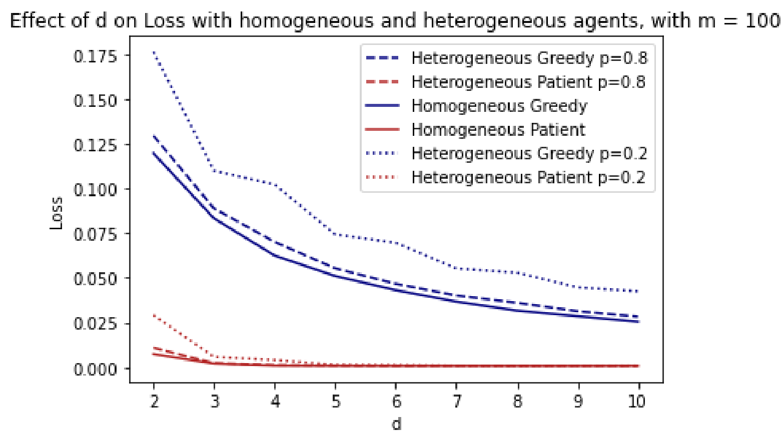

Figure 3: Loss of Greedy and Patient matching algorithms when agents that arrive at the market are either entirely homogeneous, or split into two groups of easy-to-match agents or hard-to-match agents with different probabilities. The probability p denotes the proportion of agents that are easy to match.

as agents that find a match other agents independently with probability $0.5 * \frac{d}{m}$ and easy-to-match agents as agents that find a match with other agents independently with probability $\frac{d}{m}$.

Based on our findings in Figure 4, we unsurprisingly find that the heterogeneous settings have a greater loss than the homogeneous setting, especially when there is a greater proportion of hard-to-match agents in the system. However, as d increases, we see that the loss stabilizes and the overall difference between the losses in different markets becomes less.

It appears that the Patient matching algorithm converges much sooner at greater values of d than the Greedy matching algorithm. This can likely be attributed to how the Patient matching algorithm allows the market to thicken, thus increasing the likelihood of a match for hard-to-match agents in the system. As we do not factor any waiting cost into the design, these values are not necessarily a fair representation of the overall loss in a more realistic setting.

6.3 Modifications to the Greedy Matching Algorithm - "Randomized" Greedy

To explore a modification to the baseline Greedy matching algorithm, we provide an additional parameter r to the Greedy algorithm. At every time step $t_i \in T$ the planner first matches all incoming agents but with probability r they attempt to match all critical agents at the time-step t_i, randomly mimicking what a Patient algorithm would do at that time step instead. The motivation for this is to attempt to match critical agents that have had a longer sojourn, and give them priority to find a matching before they perish.

Our results for this can be seen in Figure 3, though it is hard to make out a difference in the loss. We find that for each value of d, the "Randomized" Greedy algorithm shows a $\pm 2\%$ change in loss when compared to the standard Greedy algorithm.

While we have only looked at one setting for the randomized parameter ($r = 0.2$), we think that this naïve implementation of a strategy switching algorithm would not yield great reductions in the loss. Instead, we could try and determine periods of time where we may want to use a Patient matching strategy for a certain number of time-steps, and then switch back to the regular Greedy matching algorithm.

One limitation of this "Randomized" strategy however, is that it assumes full knowledge of when agents become critical. As mentioned briefly earlier, this is not always an assumption that one can always make, and our model may need to predict when an agent becomes critical or use a separate random variable to infer this. Doing so would add more variability to our model however, and this is a relaxation we decided to make to test our switching strategy.

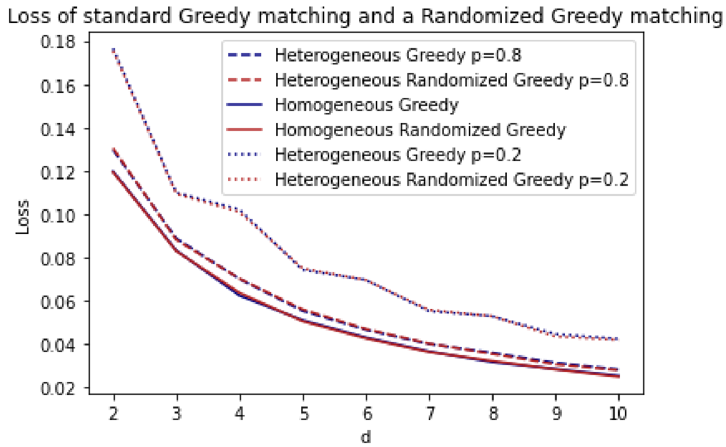

Figure 4: Loss of standard Greedy matching algorithm (blue) compared to the loss of our "Randomized" Greedy matching algorithm (red) with parameter $r = 0.2$. The difference in loss is minimal of appx. ± 1-2%

6.4 Summary

To summarize our results, we find:

- When the market is non-homogeneous, there are greater costs to the loss of matching algorithms
- Randomly choosing to match critical agents before they perish has minimal improvements on the loss when compared to baseline Greedy models
- More extensive testing is required to better understand how randomly matching agents greedily vs. patiently influences the overall loss of a matching algorithm

7 Future Work

7.1 Testing

In the future, it would likely be useful to run more tests on our model to determine the effect of other variables on the loss functions of both algorithms. In particular, we kept $\delta = 0$, based on the assumptions in Akbarpour et al. (2014), but it would be useful in the future to determine the impact of δ on the loss function. Additionally, we would like to further investigate the issue where our baseline tests with the Patient and Greedy strategies did not fit within the theoretical bounds provided (as seen in Figures 1 and 2).

7.2 Varied Compatibility

Though we present a good baseline model for dynamic matching algorithms, it is still a rather basic model and may benefit in the future from more detailed features. One example of this is that in our model, as in the model presented in Akbarpour et al. (2014), any two agents are labeled either as compatible or not compatible. That is, all compatible pairs are equally good fits for each other, and there is nothing in place to reflect different levels of compatibility. Hence, if an agent can be matched as soon as it enters the market, it will be matched (at least in a greedy market), and this presents a higher utility than if the agent had waited for a better match.

A more advanced and accurate future model may have some mechanism accounting for differences in compatibility among pairs of agents, which would assign different utilities to different pairs of agents based on these differences. Further, it may be interesting to study whether this has an impact on the trade-offs between greedy and patient algorithms, and how the weights of matches may influence the creation of chains in a matching.

7.3 Switching Strategies

One of the initial aims of this project was to determine the effect of switching from one strategy to another in a given model. While we have included a mechanism for switching strategies, we have not yet implemented a way for the market to determine when it thinks the best time for switching may be. In future iterations, it may be useful to include some variable that causes the market to switch strategies once it reaches some threshold. Examples of this threshold could be the net utility (cost) of agents in the market reaching a threshold, where it then becomes less important to thicken the market and more important to greedily match incoming agents, and vice-versa.

7.4 Matching Algorithms

A limitation of our current framework is that we use very basic matching algorithms. Instead, we could investigate more sophisticated matching strategies such as chain matching or bilateral matching, as described in Ashlagi et al. (2017), which may be more effective in heterogeneous dynamic markets. This may also give a more robust model for agent matching when combined with varied compatibility, which would lead to more realistic results from these simulations. In summary, what we have written is a baseline model based on the model in Akbarpour et al. (2020), but there are many features, both mentioned and not mentioned in this paper, which would make it a more realistic model in future iterations.

References

Mohammad Akbarpour, Shengwu Li, and Shayan Oveis Gharan. Dynamic matching market design. *arXiv preprint arXiv:1402.3643*, 2014.

Mohammad Akbarpour, Shengwu Li, and Shayan Oveis Gharan. Thickness and information in dynamic matching markets. *Journal of Political Economy*, 128(3):783–815, 2020. doi: 10.1086/704761. URL https://doi.org/10.1086/704761.

Itai Ashlagi, Maximillien Burq, Patrick Jaillet, and Vahideh Manshadi. On matching and thickness in heterogeneous dynamic markets, 2017.

Paul Erdős, Alfréd Rényi, et al. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5(1):17–60, 1960.

Alvin E. Roth, Tayfun Sönmez, and M. Utku Ünver. Efficient kidney exchange: Coincidence of wants in markets with compatibility-based preferences. *American Economic Review*, 97(3):828–851, June 2007. doi: 10.1257/aer.97.3.828. URL https://www.aeaweb.org/articles?id=10.1257/aer.97.3.828.

Erhun Özkan and Amy R. Ward. Dynamic matching for real-time ride sharing. *Stochastic Systems*, 10(1):29–70, 2020. doi: 10.1287/stsy.2019.0037. URL https://doi.org/10.1287/stsy.2019.0037.