Thickness and Information in Dynamic Matching Markets

Mohammad Akbarpour

Stanford University

Shengwu Li

Harvard University

Shayan Oveis Gharan

University of Washington

We introduce a simple model of dynamic matching in networked markets, where agents arrive and depart stochastically and the composition of the trade network depends endogenously on the matching algorithm. If the planner can identify agents who are about to depart, then waiting to thicken the market substantially reduces the fraction of unmatched agents. If not, then matching agents greedily is close to optimal. We specify conditions under which local algorithms that choose the right time to match agents, but do not exploit the global network structure, are close to optimal. Finally, we consider a setting where agents have private information about their departure times and design a mechanism to elicit this information.

This paper has been previously distributed with the title “Dynamic Matching Market Design.” We thank Paul Milgrom, Alvin Roth, and several anonymous referees for valuable guidance. We also thank Eric Budish, Ben Golub, Scott Kominers, Alex Wolitzky, and multiple anonymous referees for their insights and advice. We are also grateful to Daron Acemoglu, Nikhil Agarwal, Nick Arnosti, Itai Ashlagi, Peter Biro, Timothy Bresnahan, Jeremy Bulow, Gabriel Carroll, Sylvain Chassang, Constantinos Daskalakis, Darrell Duffie, Alex Frankel, Matthew Jackson, Fuhito Kojima, Soohyung Lee, Jacob Leshno, Malwina Luczak, Joshua Mollner, Markus Mobius, Roger Myerson, Stephen Nei, Muriel Niederle, Afshin Nikzad,

Electronically published January 30, 2020
[*Journal of Political Economy*, 2020, vol. 128, no. 3]

I. Introduction

We study the problem of matching in a dynamic market with network constraints. In many markets, only some pairs of agents can be feasibly matched. For instance, in carpooling platforms, whether two riders can share a ride depends on their locations and destinations. In paired kidney exchange, patient-donor pairs must be biologically compatible before a swap can be made. Because of these frictions, any matching decision is constrained by a network, comprised of agents (nodes) and compatible pairs (links).

In many such markets, the set of compatible agents is not fixed. Instead, agents arrive and depart over time, and a social planner continually observes the network of compatible agents and chooses how to match them. Matched agents leave the market, and unmatched agents either persist or depart. Consequently, the planner's decision today affects the sets of agents and options tomorrow. For instance, in carpooling platforms, the compatibility network evolves when new ride requests arrive, when two riders are matched, or when unmatched riders leave the market.

In such environments, the planner must decide not only which agents to match but also when to match them. The planner could match agents frequently or wait to thicken the market. If the planner waits, agents may depart. However, waiting has benefits. For example, in figure 1*A*, where each node represents an agent and each link represents a compatible pair, if the planner matches agent 1 to agent 2 at time t, then the planner will be unable to match agents 3 and 4 at time $t + 1$. By contrast, if the planner waits until $t + 1$, he can match all four agents by matching 1 to 4 and 2 to 3. Moreover, waiting might bring information about which agents will soon depart, enabling the planner to give priority to those agents. For example, in figure 1*B*, the planner learns at $t + 1$ that agent 3 will imminently leave the market if not matched. If the planner matches agent 1 to agent 2 at time t, then he will be unable to react to this information at $t + 1$.

The optimal timing policy in a dynamic matching problem is not obvious a priori. Ride-sharing platforms, for instance, have been extensively experimenting with their timing policies to analyze the trade-off between matching frequency and market thickness.[1] Many paired kidney exchanges enact static matching algorithms ("match-runs") at fixed intervals. Even

Michael Ostrovsky, Parag Pathak, Robert Shimer, Takuo Sugaya, Bob Wilson, and several anonymous referees for their valuable comments. All errors remain our own.

[1] For instance, Uber's website writes, "In the early days, a rider was immediately matched with the closest available driver. . . . But if we wait just a few seconds after a request, it can make a big difference. It's enough time for a batch of potential rider-driver matches to accumulate" (https://marketplace.uber.com/matching, accessed Dec. 2, 2018). Also, see Yan et al. (2019) for an introduction to the "dynamic waiting" algorithm, a recent innovation in timing policy at Uber.

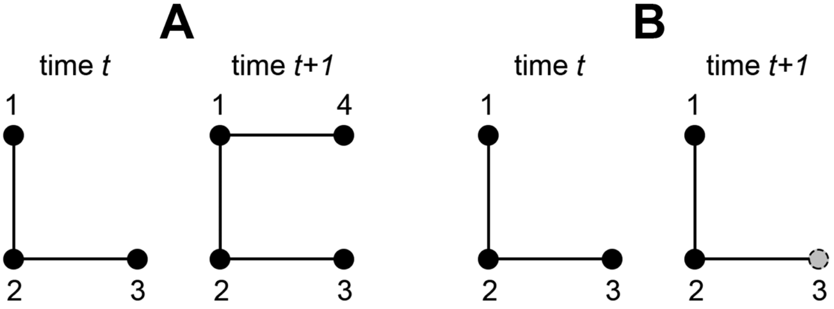

FIG. 1.—Waiting expands information about the set of options and departure times. Here, each node represents an agent and each link represents a compatible pair. In *A*, the planner observes the set of new agents and options at time $t + 1$. If he matches agent 1 to agent 2 at time t, then the planner will be unable to match agents 3 and 4 at time $t + 1$. In *B*, the planner gets the information that agent 3 is about to depart at time $t + 1$. If he matches 1 and 2 at time t, then he will be unable to react to the information about the urgency of agent 3 at time $t + 1$.

then, matching intervals differ substantially between exchanges.[2] Under what conditions is it valuable to wait to thicken the market?

We now introduce our model. In the classic Erdős-Rényi random-graph setting, there are m agents, and any two agents are compatible with probability d/m, where d scales the expected degree of each agent (Erdős and Rényi 1960). A planner observes the network and chooses a matching, seeking to minimize the number of unmatched agents. We create a natural dynamic analogue: agents arrive at Poisson rate m, any two agents are compatible with probability d/m, and each agent departs (*perishes*) at a Poisson rate, normalized to 1. Links persist over time. The planner observes the current network and chooses a matching; matched agents leave the market. The planner also observes which agents are *critical*, in the sense that he knows which agents will perish imminently if not matched. As in the static case, the planner seeks to minimize the proportion of unmatched agents (the *loss*).

In essence, our model is about matching agents to carry out some valuable activity in pairs. Agents gradually arrive and depart, and only certain pairs of agents can cooperate. The model can be interpreted as the problem faced by a ride-sharing firm that offers a carpooling service. With the pricing scheme held fixed, customers arrive and depart over time, and two riders are compatible if they have nearby locations and

[2] The Alliance for Paired Kidney Donation conducts a match-run every weekday (APKD 2017), the United Network for Organ Sharing conducts a match-run twice a week (UNOS 2015), the South Korean kidney exchange conducts a match-run once a month, and the Dutch kidney exchange conducts a match-run once a quarter (Akkina et al. 2011).

destinations. It can also be interpreted as a stylized representation of barter exchange; a pair is compatible if they can gainfully trade with each other. Kidney exchange is one example of such a trade, although the abstractions in the model are particularly aggressive for this application.[3]

What are the key features of the optimal dynamic matching algorithm? Since we explicitly model the network of potential matches, the resulting Markov decision problem is combinatorially complex. Thus, it is not feasible to compute the optimal solution with standard dynamic programming techniques. To address this issue, we employ a different approach: we formulate simple algorithms with different timing properties, which are analytically tractable because they naïvely ignore the network structure. By comparing these algorithms, we show that the choice of when to match agents has large effects on performance. Then, we produce theoretical bounds on the performance of optimal algorithms that additionally exploit the network structure. We show that, when d is not too small, our simple algorithms come close to these bounds on optimum performance. This suggests that timing is a first-order concern in our environment.

The simple algorithms are as follows. The *Greedy* algorithm attempts to match agents as soon as possible; it treats each instant as a static matching problem without regard for the future. The *Patient* algorithm attempts to match only critical agents (potentially to a noncritical partner). Both these algorithms are *local*, in the sense that they look at only the immediate neighbors of the agent they attempt to match, rather than at the global network structure.

It is intuitive that the Patient algorithm will achieve a lower loss than the Greedy algorithm, but is the difference substantial? Our first result answers this question: the Greedy algorithm's loss is at least $1/(2d + 1)$, whereas the Patient algorithm's loss is at most $e^{-d/2}/2$. To place these results in context, the static model provides a useful benchmark. Given a maximum matching on an Erdős-Rényi random graph, the expected fraction of unmatched agents is exponentially small in d, so the loss falls rapidly as d rises (Zhou and Ou-Yang 2003). In the case with arrivals and departures, our result shows that running a statically optimal matching at every instant does not yield exponentially small loss. However, waiting to match agents suffices to achieve exponentially small loss, and thus the Patient algorithm substantially outperforms the Greedy. For instance, in a market where $d = 8$, the loss of the Patient algorithm is no more than 16% of the loss of the Greedy algorithm.

The intuition behind this result is as follows. The composition of the network, as well as the number of agents in the market, depends endogenously

[3] In particular, while some agents in the model have more links than others, the Erdős-Rényi structure rules out the possibility of persistently underdemanded and overdemanded types, an important feature of kidney exchange (Roth, Sönmez, and Ünver 2005).

on the matching algorithm. As d rises, the Greedy algorithm matches agents more rapidly, reducing the equilibrium stock of available agents. This effect cancels out the exponential improvements that would accrue from raising d in a static model. In addition, under the Greedy algorithm, there are no compatible agents among the set of agents in the market (the market is thin), and so all critical agents perish. On the contrary, under the Patient algorithm, an increase in d will not rapidly reduce the equilibrium stock of available agents, so the market is always thick. This market thickness enables the planner to react to critical cases.

Our second result states that the loss of the Patient algorithm is "close to" the loss of the optimum algorithm; the optimum algorithm's loss is at least $e^{-(d/2)(1+\epsilon)}/(d + 1)$ where $\epsilon \le e^{-d/2}$. Recall that the Patient algorithm is local; it looks at only the immediate neighborhood of the agents it seeks to match. By contrast, the optimum algorithm chooses the optimal time to match agents, as well as the optimal agents to match, by exploiting the entire network structure. When we compare the performance of the Greedy algorithm to that of the optimum algorithm, we find that most of the gain is achieved merely by thickening the market, rather than by optimizing over the network structure.

Essentially, our first two results delineate conditions under which the timing of matching matters much more than optimizing over the network. They suggest that it may be unwise to abstract away from timing considerations in order to focus on the network structure.

Next, we drop the assumption that the planner can identify critical agents and ask, What if the planner has more or less information about departure times? Our next results show that departure information and market thickness are complements, in the following sense. Any algorithm that cannot identify critical agents has a loss of at least $1/(2d + 1)$, no matter how long it waits. Suppose, on the other hand, that the planner is constrained to match agents as soon as possible but knows agents' departure times far in advance. Any algorithm that does not wait has a loss of at least $1/(2d + 1)$, no matter how much information it has about departure times.

Recall that the Patient algorithm requires only short-horizon information about agent departures. What if the planner has even more information? For instance, the planner may be able to forecast departures long in advance or foresee how many new agents will arrive or may know that certain agents are more likely than others to have new links. We prove that no expansion of the planner's information allows him to achieve a loss smaller than $e^{-d}/(d + 1)$. Taken together, these results suggest that short-horizon information about departure times is especially valuable to the planner. Lacking this information leads to large losses, and having more than this information does not yield large gains.

In some settings, however, agents know that they are critical, but the planner does not. For instance, riders know whether they are in urgent need of a car or can wait with low cost, but the carpooling platform does not. Our final result concerns the incentive-compatible implementation of the Patient algorithm. Suppose that the planner observes the network but does not know when agents are critical. Suppose that agents know when they are critical but do not observe the network (i.e., they do not know when they have a compatible partner).[4] When agents have waiting costs, they may have incentives to misreport their urgency, so as to hasten their match or to increase their probability of getting matched. We show that if agents are not too impatient, a dynamic mechanism without transfers can elicit such information. The mechanism treats agents who report that they are critical but persist as though they had left the market. This means that as an agent, I trade off the possibility of a swifter match (by declaring that I am critical now) with the option value of being matched to another agent before I truly become critical. We prove that it is arbitrarily close to optimal for agents to report the truth in large markets.

We close by stating a caveat. Our results build on several stylized assumptions. Whether these are plausible depends on the context. For instance, consider the assumption of exponential departure times. In kidney exchanges, it seems implausible to assume that a patient's departure time is memoryless, but this assumption is consistent with empirical evidence of ride sharing (Liu, Wan, and Yang 2019). Hence, our results do not imply that network optimization is never important. Rather, they specify conditions under which timing considerations are more important than network optimization. If an analyst believes (or finds out) that employing a complicated algorithm that accounts for the network structure will yield large gains, then their environment must depart materially from the setting studied here.

Related work.—There have been several studies on dynamic matching in economics, computer science, and operations research. To the best of our knowledge, no prior work has examined dynamic matching on a general random graph, where agents stochastically depart.

Kurino (2009) and Bloch and Houy (2012) study an overlapping-generations model of the housing market. In their models, agents have deterministic arrivals and departures, and the housing side of the market is infinitely durable and static. In the same context, Leshno (2012) studies a one-sided dynamic housing allocation problem in which there are two types of houses that arrive stochastically over time. In subsequent papers, Baccara, Lee, and Yariv (2015) and Loertscher, Muir, and Taylor (2016) study the problem of optimal dynamic matching and thickness in two-sided models with two types on each side.

[4] One of the reasons that agents enter centralized matching markets is that they are unable to find partners by themselves.

In the context of kidney exchanges, a problem first studied by Roth, Sönmez, and Ünver (2004, 2005), Ünver (2010) is the first paper that considers dynamics in a model with multiple types of agents. In Ünver's model, agents never perish. Thus, one insight of his model is that waiting to thicken the market is not helpful when only bilateral exchanges are allowed. We show that this result changes when agents depart stochastically. Some other aspects of dynamic kidney exchange have been studied by Zenios (2002), Su and Zenios (2005), Awasthi and Sandholm (2009), Dickerson, Procaccia, and Sandholm (2012), and Sonmez and Ünver (2015). Ashlagi, Jaillet, and Manshadi (2013) construct a finite-horizon model of kidney exchange with agents who never depart. They show that (with two-way exchanges) waiting yields large gains only if the planner waits for a constant fraction of total agents to arrive. Since our model is infinite horizon and agents depart, it is not possible to wait for a constant fraction of the total agents to arrive. Nevertheless, the Patient algorithm ensures that the size of the market is linear in m, which makes thickness valuable. A recent paper builds on our framework to study the competition of two platforms with Greedy and Patient algorithms (Das et al. 2015).

In concurrent work, Anderson et al. (2015) analyze a model in which the main objective is to minimize the average waiting time and agents never perish. They show that with two-way exchanges, the Greedy algorithm is optimal in the class of "periodic Markov policies," which is similar to our theorem 4. Our paper shows that when agents' departure times are observable, Greedy performs weakly, and the option value of waiting can be large. In other related work, Arnosti, Johari, and Kanoria (2014) model a two-sided dynamic matching market to analyze congestion in decentralized markets. Some recent papers study the problem of stability in dynamic matching markets (Du and Livne 2014; Kadam and Kotowski 2018; Doval 2016). We are concerned with total welfare, as opposed to stability.

In recent years, research on ride-sharing platforms has grown rapidly. While the problem of pricing is widely studied, the literature on the problem of dynamic matching in ride sharing is relatively sparse. For instance, Özkan and Ward (2017) and Ashlagi et al. (2018) studied this problem from a theoretical perspective. More recently, Liu, Wan, and Yang (2019) develop a two-sided version of our model and estimate it using data from DiDi, the world's largest ride-sharing platform. Their comparisons of the Patient and Greedy algorithms show that if drivers are not too heterogeneous, Patient outperforms Greedy.

The literature on online advertising is also related to our work. In this setting, advertisements are static, but queries arrive adversarially or stochastically over time. Unlike our model, queries persist in the market for exactly one period. Karp, Vazirani, and Vazirani (1990) introduced the problem and designed a randomized matching algorithm. Subsequently, the problem has been considered under several variations; see, for instance, Mehta et al. (2007), Goel and Mehta (2008), Manshadi, Gharan, and Saberi (2012), and Blum et al. (2015).

The problem of dynamic matching has been extensively studied in the literature on labor market search. Shimer and Smith (2001) study a decentralized search market and discuss efficiency issues. This paper and its descendants are different from ours in at least two ways. First, rather than modeling market thickness via a fixed match function, we explicitly account for the network structure that affects the planner's options, endogenously determining market thickness. In addition, in Shimer and Smith (2001), the benefit of waiting is in increasing the match quality, whereas in our model we show that even if you cannot increase match quality, waiting can still be beneficial because it increases the number of agents who get matched. Ebrahimy and Shimer (2010) study a decentralized version of the Greedy algorithm from a labor-search perspective.[5]

II. Model

In this section, we introduce the pieces of our continuous-time model for a matching market on stochastic networks that runs in the interval $[0, T]$.

A. *Arrivals and Departures*

Agents arrive at the market at Poisson rate m. Hence, in any interval $[t, t+1]$, m new agents enter the market in expectation. Throughout the paper, we assume $m \geq 1$. Let A_t be the set of the agents in our market at time t, and let $Z_t := |A_t|$. We refer to A_t as the *pool* of the market and to Z_t as the *pool size*. We start by describing the evolution of A_t as a function of $t \in [0, T]$. Since we are interested in the limit behavior of A_t, we assume $A_0 = \varnothing$. We use A_t^{n} to denote the set of agents who enter the market at time t.[6] Note that with probability 1, $|A_t^{\mathrm{n}}| \leq 1$. Also, let A_{t_0,t_1}^{n} denote the set of agents who enter the market in time interval $[t_0, t_1]$. Each agent becomes critical according to an independent Poisson process at rate λ, which, without loss of generality, we normalize to 1. This implies that, if an agent a enters the market at time t_0, then she becomes critical at some time $t_0 + X$, where X is an exponential random variable with mean 1. Any critical agent leaves the market immediately; so the last point in time that an agent

[5] A less directly related literature is that of dynamic auctions and dynamic mechanism design. Budish, Cramton, and Shim (2015) study the problem of timing and frequent batch auctions in the high-frequency setting. Parkes and Singh (2003) generalize the Vickrey-Clarke-Groves mechanism to a dynamic setting. We refer interested readers to Parkes (2007) for a review of the dynamic mechanism design literature.

[6] As a notational guidance, we use subscripts to refer to a point in time or a time interval, while superscripts "n" and "c" refer to new agents and critical agents, respectively.

can get matched is the time that she becomes critical. We say an agent a perishes if a leaves the market unmatched.

We assume that an agent $a \in A_t$ leaves the market at time t if a is not critical but is matched with another agent $b \in A_t$, if a becomes critical and gets matched to another agent, or if a becomes critical and leaves the market unmatched and so perishes. Consequently, for any matching algorithm, a leaves the pool at some time t_1 where $t_0 \le t_1 \le t_0 + X$. The *sojourn* of a is the length of the interval that a is in the pool, that is, $s(a) := t_1 - t_0$. We use A_t^c to denote the set of agents who are critical at time t.[7] Also, note that for any $t \ge 0$, with probability 1, $|A_t^c| \le 1$.

It is essential to note that the arrival of the criticality event with some Poisson rate is not equivalent to discounting with the same rate, because the criticality event might be observed by the planner and the planner can react to that information.

B. The Compatibility Network

For any pair of agents, they are compatible with probability p, where $0 \le p \le 1$, and these probabilities are independent across pairs. Let $d = m \cdot p$ be the density parameter of the model. In the paper, we use this definition and replace p with d/m.

For any $t \ge 0$, let $E_t \subseteq A_t \times A_t$ be the set of compatible pairs of agents in the market (the set of *edges*) at time t, and let $G_t = (A_t, E_t)$ be the network at time t. Compatible pairs persist over time; that is, if $a, b \in A_t$ and $a, b \in A_{t'}$, then $(a, b) \in E_t$ if and only if $(a, b) \in E_{t'}$. For an agent $a \in A_t$, we use $N_t(a) \subseteq A_t$ to denote the set of neighbors of a in G_t. It follows that, if the planner does not match any agents, then for any fixed $t \ge 0$, G_t is distributed as an Erdős-Rényi graph with parameter d/m and, in the long run, d is the average degree of agents (Erdős and Rényi 1960).[8]

Let $A = \cup_{t \le T} A_t^n$, let $E \subseteq A \times A$ be the set of acceptable transactions between agents in A, and let $G = (A, E)$.[9] Observe that any realization of the above stochastic process is uniquely defined, given $A_t^{\mathbf{n}}$, A_t^c for all $t \ge 0$ and the set of compatible pairs, E. A vector (m, d) represents a *dynamic matching market.*

[7] In our proofs, we use the fact that $A_t^c \subseteq \cup_{0 \le \tau \le t} A_\tau$. In the example of the text, we have $a \in A_{t_0+X}^c$. Note that even if agent a is matched before becoming critical (i.e., $t_1 < t_0 + X$), we still have that $a \in A_{t_0+X}^c$. Hence, A_t^c is not necessarily a subset of A_t, since it may have agents who are already matched and have left the pool. This generalized definition of A_t^c is helpful in our proofs.

[8] In an undirected graph, the *degree* of a vertex is equal to the total number of edges connected to that vertex.

[9] Note that $E \supseteq \cup_{t \le T} E_t$ and the two sets are not typically equal, since two agents may find it acceptable to transact, even though they are not in the pool at the same time because one of them was matched earlier.

C. *Matching Algorithms*

A set of edges $M_t \subseteq E_t$ is a *matching* if no two edges share the same end points. A *matching algorithm*, at any time $t \geq 0$, selects a (possibly empty) matching, M_t, in the current graph G_t, and the end points of the edges in M_t leave the market immediately. We assume that any matching algorithm at any time t_0 knows the current graph G_t only for $t \leq t_0$ and does not know anything about G_t for $t' > t_0$. In the benchmark case that we consider, the matching algorithm can depend on the set of critical agents at time t. Nonetheless, we extend several of our theorems to the case where the algorithm knows more than this or less than this.

We emphasize that the random sets A_t (the set of agents in the pool at time t), E_t (the set of compatible pairs of agents at time t), and $N_t(a)$ (the set of an agent a's neighbors), as well as the random variable Z_t (pool size at time t), are all functions of the underlying matching algorithm. We abuse notation and do not include the name of the algorithm when we analyze these variables.

D. *The Goal*

Let $\mathrm{ALG}(T)$ be the set of matched agents by time T,

$$\mathrm{ALG}(T) \coloneqq \{a \in A : a \text{ is matched by ALG by time } T\}.$$

We may drop the T in the notation $\mathrm{ALG}(T)$ if it is clear from context.

The goal of the planner is to match the maximum number of agents or, equivalently, to minimize the number of perished agents. The loss of a matching algorithm ALG is defined as the ratio of the expected number of perished agents to the expected number of agents, which is, by definition, a number in [0, 1]:

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

The planner seeks a maximum matching in a dynamic random graph. Unlike the static problem, the planner faces two additional constraints: first, not all agents are present at the same time, and second, he is uncertain about future arrivals and departures.

Minimizing loss is equivalent to maximizing social welfare, for the case where the cost of waiting is negligible compared to the cost of leaving the market unmatched.[10]

[10] The case where discount rate is not zero is extensively studied in a working paper (Akbarpour and Li 2019).

Our problem can be modeled as a Markov decision problem (MDP) that is defined as follows. The state space is the set of pairs (H, B) where H is any undirected graph of any size and, if the algorithm knows the set of critical agents, B is a set of at most one vertex of H representing the corresponding critical agent. The action space for a given state is the set of matchings on the graph H. Under this view, an algorithm designer wants to minimize the loss over a time period T.

E. *Optimum Solutions*

In many parts of this paper we compare the performance of a matching algorithm to the performance of an optimal Omniscient algorithm. Unlike any matching algorithm, the Omniscient algorithm has full information about the future; that is, it knows the full realization of the graph G of agents who will be in the system in future and the edges between them. Therefore, it can return the (static) maximum matching in this graph as its output and thus minimize the fraction of perished agents. Let $\mathrm{OMN}(T)$ be the set of matched agents in the maximum matching of G. The loss function under the Omnsicient algorithm at time T is

$$\mathbf{L}(\mathrm{OMN}) \coloneqq \frac{\mathbb{E}[|A - \mathrm{OMN}(T) - A_T|]}{mT}.$$

Observe that for any matching algorithm ALG and any realization of the probability space, we have $|\mathrm{ALG}(T)| \leq |\mathrm{OMN}(T)|$, because the Omniscient algorithm knows the realization of the stochastic process and could select the same matching as any matching algorithm.

The optimum matching algorithm, that is, the solution to the above MDP, is the algorithm that minimizes loss. We first consider OPT^{c}, the algorithm that knows the set of critical agents at time t. We then relax this assumption and consider OPT, the algorithm that does not know these sets.

Let ALG^{c} be any matching algorithm that knows the set of critical agents at time t. It follows that

$$\mathbf{L}(\mathrm{ALG}^{c}) \geq \mathbf{L}(\mathrm{OPT}^{c}) \geq \mathbf{L}(\mathrm{OMN}).$$

Similarly, let ALG be any matching algorithm that does not know the set of critical agents at time t. It follows that

$$\mathbf{L}(\mathrm{ALG}) \geq \mathbf{L}(\mathrm{OPT}) \geq \mathbf{L}(\mathrm{OPT}^{c}) \geq \mathbf{L}(\mathrm{OMN}).$$

Note that unlike the loss function that considers the expected number of matched agents, $|\mathrm{ALG}|$ and $|\mathrm{OPT}|$ (the number of matched agents under ALG and OPT) are generally incomparable, and, depending on the realization of G, we may even have $|\mathrm{ALG}| > |\mathrm{OPT}|$.

III. Simple Matching Algorithms

In our model, solving for the optimal matching algorithm is computationally complex. This is because there are at least $2^{\binom{m}{2}}/m!$ distinct graphs of size m, so for even moderately large markets, we cannot apply standard dynamic programming techniques to find the optimum online matching algorithm.[11]

Nevertheless, we are not fully agnostic about the optimal algorithm. In particular, we know that OPT^c has at least two properties:

i. A pair of agents a, b gets matched in OPT^c only if one of them is critical, because if a, b can be matched and neither of them is critical, then we are weakly better off if we wait and match them later.
ii. If an agent a is critical at time t and $N_t(a) \neq \varnothing$, then OPT^c matches a. This is because allowing a critical agent to perish now decreases the number of future perished agents by at most 1.

OPT^c waits until some agent becomes critical, and if an agent is critical and has some compatible partner, then OPT^c matches that agent. But the choice of match partner depends on the entire network structure, which is what makes the problem combinatorially complex. Our goal here is to separate these two effects. How much is achieved merely by being patient? And how much more is achieved by optimizing over the network structure?

To do this, we start by designing a matching algorithm (the Greedy algorithm) that mimics “match-as-you-go” algorithms used in many real marketplaces. It delivers maximal matchings at any point in time, without regard for the future.

Definition 1 (Greedy algorithm). If any new agent a enters the market at time t, then match her with an arbitrary agent in $N_t(a)$ whenever $N_t(a) \neq \varnothing$.

Since no two agents arrive at the same time almost surely, we do not need to consider the case where more than one agent enters the market. Moreover, the graph G_t in the Greedy algorithm is almost always an empty graph. Hence, the Greedy algorithm cannot use any information about the set of critical agents.

To separate the value of waiting from the value of optimizing over the network structure, we design a second algorithm, which chooses the optimal time to match agents but ignores the network structure.

[11] This lower bound is derived as follows. When there are m agents, there are (m_2) possible edges, each of which may be present or absent. Some of these graphs may have the same structure but different agent indices. A conservative lower bound is to divide by all possible relabelings of the agents $(m!)$. For instance, for $m = 30$, there are more than 10^{98} states in the approximated MDP.

Definition 2 (Patient algorithm). If an agent a becomes critical at time t, then match her uniformly at random with an agent in $N_t(a)$ whenever $N_t(a) \neq \emptyset$.

To run the Patient algorithm, we need access to the set of critical agents at time t. Note that the Patient algorithm exploits only short-horizon information about critical agents, as compared to the Omniscient algorithm, which has full information about the future. Of course, the knowledge of the exact departure times is an abstraction from reality, and we do not intend the timing assumptions about critical agents to be interpreted literally. An agent's point of perishing represents the point at which it ceases to be socially valuable to match that agent, and the agent will incur a high waiting cost if not matched. Letting the planner observe the set of critical agents is a modeling convention that represents high-accuracy short-horizon information about agents' departures.

We now state results for the case of large markets with sparse graphs, in the steady state: $m \to \infty$, d is held constant, and T goes to infinity at a rate faster than $\log(m)$. Clearly, this implies that $d/m = p \to 0$, which should not be taken literally. This method eliminates nuisance terms and is a standard way to state results for large but sparse graphs (Erdős and Rényi 1960). Appendix A studies the performance of each algorithm as a function of m, T, and d, without taking limits. Simulations in online appendix G indicate that the key comparisons hold for small values of m and T. Moreover, the algorithms we examine converge rapidly to the stationary distribution, as shown in online appendix A. Readers interested in technical nonlimit results can see theorems 6 and 7 for the exact dependency of our results on m and T.

IV. Timing, Thickness, and Network Optimization

A. Theorems

Does timing substantially affect the performance of dynamic matching algorithms? Our first result establishes that varying the timing properties of simple algorithms has large effects on their performance.

Theorem 1. For $d \geq 2$, as $T, m \to \infty$,

$$\mathbf{L}(\text{Greedy}) \geq \frac{1}{2d+1},$$

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

We already knew that the Patient algorithm outperforms the Greedy algorithm. What this theorem shows is that the Patient algorithm achieves exponentially small loss, but the Greedy algorithm does not. Theorem 1 provides an upper bound for the value of waiting. We shut down the channels by which waiting can be costly (waiting cost is negligible, and the planner observes critical agents) and show that in this world, the option value of waiting is large.

Why does this happen? The Greedy algorithm attempts to match agents upon arrival, and the Patient algorithm attempts to match them upon departure. Now, suppose that a new agent enters the market under the Greedy algorithm. If there are z other agents in the market, the probability that the agent has no feasible partner is $[1 - (d/m)]^z$, which falls exponentially as d rises. Equally, suppose that an agent becomes critical and the Patient algorithm attempts to match him. If there are z other agents in the market, the probability that the agent has no feasible partner is again $[1 - (d/m)]^z$. What, then, explains the difference in performance?

The key is to see that the composition and number of agents in the market depend endogenously on the matching algorithm. As d rises, the Greedy algorithm matches agents more rapidly, depleting the stock of available agents and reducing the equilibrium z. This effect cancels out the exponential improvements that would accrue from raising d in a static model. By contrast, because the Patient algorithm waits, the market is thick. We prove that, under the Patient algorithm, equilibrium z is always above $m/2$, which entails that $[1 - (d/m)]^z$ falls exponentially as d rises.

The next question is, Are the gains from waiting large, compared to the total gains from optimizing over the network structure? First, we show by example that the Patient algorithm is not optimal, because it ignores the global network structure.

Example 1. Let G_t be the graph shown in figure 2, and let $a_2 \in A_t^c$, that is, a_2 is critical at time t. Observe that it is strictly better to match a_2 to a_1, as opposed to a_3. Nevertheless, since the Patient algorithm makes decisions that depend only on the immediate neighbors of the agent it is trying to match, it cannot differentiate between a_1 and a_3 and will choose either of them with equal probability.

The next theorem provides a bound for gains from network optimizations by proving a lower bound for the optimum.

Theorem 2. Consider any algorithm ALG that observes the set of critical agents. Then, for $d \geq 2$, as $T, m \to \infty$,

$$\mathbf{L}(\mathrm{OPT}^c) \geq \frac{e^{-(d/2)(1+\mathbf{L}(\mathrm{ALG}))}}{d+1}.$$

Recall that $\mathbf{L}(\mathrm{Patient}) \leq (1/2) \cdot e^{-d/2}$. Substituting for $\mathbf{L}(\mathrm{ALG})$ implies that $\mathbf{L}(\mathrm{OPT}^c) \geq e^{-(d/2)(1+e^{-d/2}/2)}/(d+1)$.[12] This exponential term in $\mathbf{L}(\mathrm{OPT}^c)$

[12] Paradoxically, this argument exploits the fact that Patient has a small loss to show that OPT^c has a large loss. The bound in theorem 2 is decreasing $\mathbf{L}(\mathrm{ALG})$, so new algorithms that improve on Patient automatically result in a tighter lower bound on $\mathbf{L}(\mathrm{OPT}^c)$.

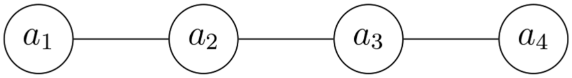

FIG. 2.—If a_2 becomes critical in this graph, it is strictly better to match him to a_1, as opposed to a_3. The Patient algorithm, however, chooses either a_1 or a_3 with equal probability.

is close to that of the **L**(Patient) for even moderate values of d. The preceding results show that the gains from the right timing decision (moving from the Greedy algorithm to the Patient algorithm) are larger than the remaining gains from optimizing over the entire network (moving from the Patient algorithm to the optimum algorithm). In many settings, optimal solutions may be computationally demanding and difficult to implement. Thus, this result suggests that, in settings that approximately satisfy our key assumptions, it will often be more worthwhile for policy makers to find ways to thicken the market than to seek complicated optimal policies.

It is worth emphasizing that this result (as well as theorem 4) proves that "local" algorithms are close to optimal. Since in our model agents are ex ante homogeneous, this shows that "whom to match" is not as important as "when to match." In settings where agents have multiple types, however, the decision of "whom to match" can be an important one even when it is local. We defer the discussion of conditions under which "whom to match" is important to section VI.

B. *Proof Overview*

We now sketch the proof and offer intuition for theorem 1. The proof of theorem 2 has ideas similar to the proof of theorem 4, so we discuss them together in the next section.

The key idea in proving theorem 1 is to study the structure of the graph induced by these algorithms and the distribution of the pool size, Z_t. In particular, we show that Z_t is a Markov chain with a unique stationary distribution that mixes rapidly and that it is a sufficient statistic for the structure of the graph under the Greedy and Patient algorithms.

Greedy algorithm.—Under the Greedy algorithm, conditional on Z_t, the pool is almost always an empty graph, that is, a graph with no edges. Now note that the rate at which some agent in the pool becomes critical is Z_t. Because the graph is empty, critical agents perish with probability 1. Therefore, in steady state, $\mathbf{L}(\text{Greedy}) \approx \mathbb{E}[Z_t]/m$.

Next, we show that for the Greedy algorithm, $\mathbb{E}[Z_t] \geq m/(2d+1)$. Take any pool size z. At rate m, a new agent arrives. With probability $(1-d/m)^z$, the new agent has no compatible matches, which increases the pool size by 1. With probability $1-(1-d/m)^z$, the new agent has a compatible match, and the pool size falls by 1. At rate z, an agent perishes, in which case the pool size falls by 1. Let z^* be the point where these forces balance, that is, the solution to

$$m(1 - d/m)^z(+1) + z(-1) + m[1 - (1 - d/m)^z](-1) = 0.$$

By algebraic manipulation, $z^* \geq m/(2d + 1)$. We show that under the stationary distribution, Z_t is highly concentrated around z^*, which then implies that $\mathbb{E}[Z_t]$ is close to z^*. This produces the lower bound for $\mathbf{L}$(Greedy).

Patient algorithm.—Under the Patient algorithm, conditional on Z_t, the pool is an Erdős-Rényi random graph with parameter d/m. To see why, suppose that an agent becomes critical. The Patient algorithm's choice of a match partner for that agent depends on only the immediate neighbors of that agent. Consequently, after the critical agent leaves, the rest of the graph is still distributed as an Erdős-Rényi random graph. The rate at which some agent becomes critical is Z_t. Because the graph is a random graph, critical agents perish with probability $(1 - d/m)^{Z_t}$. Therefore, in steady state, $\mathbf{L}(\text{Patient}) \approx \mathbb{E}[Z_t(1 - d/m)^{Z_t}]/m$.

The next step is to show that Z_t is highly concentrated around $\mathbb{E}[Z_t]$, so $\mathbf{L}(\text{Patient}) \approx \mathbb{E}[Z_t](1 - d/m)^{\mathbb{E}[Z_t]}/m$. This step involves long arguments. But once this step is established, it remains to prove that $\mathbb{E}[Z_t] \geq m/2$. The exact proof for this is involved, but a simple thought experiment gives good intuition. Suppose that the Patient algorithm is not able to match any agents. Then, $\mathbb{E}[Z_t] = m$. On the other hand, suppose that the Patient algorithm can match all agents. Then, agents arrive at rate m and get matched at rate $2\mathbb{E}[Z_t]$, because for each critical agent, two agents get matched. This implies that $\mathbb{E}[Z_t] = m/2$. In fact, the Patient algorithm can match some but not all agents, so $m/2 \leq \mathbb{E}[Z_t] \leq m$. This produces the upper bound for $\mathbf{L}$(Patient).

What if the planner has more than short-horizon information about agents' departure times? Suppose that the planner knows the exact departure times of all agents who are in the pool. Is it still the case that waiting is highly valuable? To answer this question, we design a new class of algorithms that are constrained to match agents as soon as they can but have access to exact departure times of all agents in the market. We refer to this class of algorithms as *departure-aware greedy* (DAG) algorithms.

DEFINITION 3 (DAG algorithms). If any new agent a enters the market at time t, then match her with an agent in $N_t(a)$ whenever $N_t(a) \neq \varnothing$, where the choice of match partner can depend on the profile of departure times for agents in the pool.

For instance, if a newly arrived agent has multiple matches, a DAG algorithm can break the tie in favor of the partner who departs soonest. If this algorithm can perform "close to" the Patient algorithm, then it suggests that waiting is not valuable if the planner has access to sufficiently rich information. Our next theorem, however, shows that even this long-horizon information cannot substantially help the planner.

THEOREM 3. For any DAG algorithm, as $T \to \infty$,

$$\mathbf{L}(\mathrm{DAG}) \geq \frac{1}{2d+1}.$$

Proof. Suppose that we run a DAG algorithm from time 0 to time T. Pick an agent i uniformly at random from the set of all agents who arrive between 0 and T, that is, uniformly at random from the set $A = \cup_{t \leq T} A_t^{\mathrm{n}}$. Let ϕ_i^{n} denote the probability that i is not matched upon arrival, and let ϕ_i denote the probability that i is not matched at all. Our goal is to provide a lower bound for ϕ_i.

For any realization of the stochastic process under a DAG algorithm, the number of agents who are matched is twice the number of agents who are matched upon arrival. Thus, since i was drawn uniformly at random,

$$1 - \phi_i = 2(1 - \phi_i^{\mathrm{n}}). \tag{1}$$

DAG algorithms cannot condition matching decisions on information about agents who have not yet arrived. For all other agents j, i's sojourn length is drawn independently from the probability that i is compatible with j. Thus, conditional on i not being matched on arrival, i's sojourn length is still distributed exponentially with parameter 1. New agents compatible with i arrive at Poisson rate $m \times d/m = d$. Thus, the probability that no compatible partner arrives during i's sojourn is $1/(d+1)$, so $\phi_i \geq \phi_i^{\mathrm{n}}/(d+1)$.[13] Substituting for ϕ_i^{n} with equation (1) yields $\phi_i \geq 1/(2d+1)$.

By definition of the loss,

$$\begin{aligned}\mathbf{L}(\mathrm{DAG}) &= \frac{\mathbb{E}[|A - \mathrm{DAG}(T)|]}{mT} - \frac{\mathbb{E}[|A_T|]}{mT} \geq \frac{mT[1/(2d+1)]}{mT} - \frac{m}{mT} \\ &= \frac{1}{2d+1} - \frac{1}{T}.\end{aligned} \tag{2}$$

Condition $\mathbb{E}[|A_T|] \leq m$ holds, since any algorithm has a smaller expected pool size than the inactive algorithm. Taking the limit as $T \to \infty$ completes the proof. QED

The theorem shows that any matching algorithm that does not wait, even with access to long-horizon information about departure times of agents who are in the market, cannot perform close to the Patient algorithm.

[13] For two independent Poisson arrival processes with rates 1 and d, the probability that the first process arrives before the second is $1/(d+1)$. Notably, this inequality holds regardless of the state of the system at the instant when i arrives, so the argument does not require that the system has converged to a stationary distribution (if it exists).

Therefore, the Patient algorithm strongly outperforms the Greedy algorithm because it waits long enough to create a thick market.

V. Value of Information and Incentive Compatibility

Up to this point, we have assumed that the planner knows the set of critical agents; that is, he has accurate short-horizon information about departures. We now relax this assumption in both directions.

First, we consider the case in which the planner does not know the set of critical agents. That is, the planner's policy may depend on the graph G_t but not on the set of critical agents A_t^c. Recall that OPT is the optimum algorithm subject to these constraints. Second, we consider OMN, the case under which the planner knows everything about the future realization of the market. Our main result in this section is stated below.

Theorem 4. For $d \geq 2$, as $T, m \to \infty$,

$$\frac{1}{2d+1} \leq \mathbf{L}(\mathrm{OPT}) \leq \mathbf{L}(\mathrm{Greedy}) \leq \frac{\log(2)}{d},$$
$$\frac{e^{-d}}{d+1} \leq \mathbf{L}(\mathrm{OMN}) \leq \mathbf{L}(\mathrm{Patient}) \leq \frac{1}{2} \cdot e^{-d/2}.$$

This shows that the losses of OPT and Greedy are relatively close, which indicates that waiting and criticality information are complements: waiting to thicken the market is substantially valuable only when the planner can identify critical agents. Observe that OPT could in principle wait to thicken the market, but this result proves that the gains from doing so (compared to those from running the Greedy algorithm) are not large.

What if the planner knows more than just the set of critical agents? For instance, the planner may have long-horizon forecasts of agent departure times, or the planner may know that certain agents are more likely to have matches in future than other agents.[14] However, theorem 4 shows that no expansion of the planner's information set yields a better-than-exponential loss.

Under these new information assumptions, we once more find that local algorithms can perform close to computationally intensive global optima: Greedy is close to OPT without access to criticality information, and Patient is close to OPT^c.

[14] In our model, the number of acceptable transactions that a given agent will have with the next N agents to arrive is Bernoulli distributed. If the planner knows beforehand whether a given agent's realization is above or below the 50th percentile of this distribution, it is as though agents have different "types."

We now sketch the proof of bounds that we provided for our optimum benchmarks. The full proofs can be found in appendix B, as well as in online appendix C.3.

OPT algorithm.—We show how we bound $\mathbf{L}(\text{OPT})$, without knowing anything about the way OPT works. The idea is to provide lower bounds on the performance of any matching algorithm as a function of its expected pool size. Let ζ be the expected size of the pool at time t, where t is picked uniformly at random from $[0, T]$. The rate at which some agent becomes critical is ζ. When the planner does not observe critical agents, all critical agents perish. Hence, $\mathbf{L}(\text{OPT}) = \zeta/m$. Note that this is an increasing function of ζ, so from this perspective the planner prefers to decrease the pool size as much as possible.

Next, we count the fraction of agents who do not form any edges upon arrival and during their sojourn. No matching algorithm can match these agents, and so the fraction of those agents is a lower bound on the performance of any matching algorithm, including OPT. The probability of having no edges upon arrival is at least $(1 - d/m)^{\zeta}$, while (as $T \to \infty$) the probability of not forming any edges during a sojourn is $\int_{t=0}^{\infty} e^{-t} \cdot (1 - d/m)^{mt}\, dt$, because an agent who becomes critical t periods after arrival meets mt new agents in expectation. Simple algebra (see sec. B1) shows that for any algorithm ALG,

$$\mathbf{L}(\text{ALG}) \geq \frac{e^{-\zeta(1+d/m)d/m}}{1 + d + d^2/m} \geq \frac{1 - \zeta(d/m + d^2/m^2)}{1 + 2d + d/m^2}.$$

From this perspective, the planner prefers to increase the pool size as much as possible. One can then easily show that if $\zeta \leq 1/(2d + 1)$, this lower bound guarantees that the fraction of agents with no matches is at least $1/(2d + 1)$, and if $\zeta > 1/(2d + 1)$, our previous bound guarantees that the loss is at least $1/(2d + 1)$. So $\mathbf{L}(\text{OPT}) \geq 1/(2d + 1)$.

OMN algorithm.—We use a similar trick to provide a lower bound for $\mathbf{L}(\text{OMN})$. We have already established a lower bound on the fraction of agents with no matches, as a function of expected pool size. But we know that the expected pool size can never be more than m, because that is the expected pool size when the planner does not match any agents. Hence, the fraction of agents with no matches when the expected pool size is m is a lower bound on the loss of the OMN. (See sec. B2 for the details.)

OPT^c algorithm. Now we sketch the proof of theorem 2 and bound OPT^c. The key idea behind this proof is the following: OPT^c matches agents (to noncritical partners) if and only if they are critical. Suppose that we run OPT^c from time 0 to T. Take any realization of the stochastic process: all critical agents depart. A noncritical agent departs if and only if he is matched. Thus, the number of noncritical agents who are matched is equal to the number of departures (denoted #depart) minus the number of agents who become critical (denoted #critical). Each matched pair consists of one critical agent and one noncritical agent, so the total number of matched agents is equal to 2(#depart − #critical). The following approximations hold as $T \to \infty$: $\mathbf{L}(\mathrm{OPT}^{\mathrm{c}}) \approx \mathbb{E}[\#\mathrm{arrive} - 2(\#\mathrm{depart} - \#\mathrm{critical})]/mT$, where #arrive is the number of agents who arrive between 0 and T; $\mathbb{E}[\#\mathrm{critical}] = \zeta^{\mathrm{c}}$, where ζ^{c} is the expected pool size at t for t drawn uniformly at random from $[0, T]$; and $\mathbb{E}[\#\mathrm{arrive} - 2\#\mathrm{depart}]/mT \approx -1$. Thus, $\mathbf{L}(\mathrm{OPT}^{\mathrm{c}}) \approx 2(\zeta^{c}/m) - 1$.

Finally, for any ALG, $\mathbf{L}(\mathrm{ALG}) \geq \mathbf{L}(\mathrm{OPT}^{\mathrm{c}}) \approx 2(\zeta^{\mathrm{c}}/m) - 1$, which leads to $(m/2)(1 + \mathbf{L}(\mathrm{ALG})) \geq \zeta^{\mathrm{c}}$. From above, $\mathbf{L}(\mathrm{OPT}^{\mathrm{c}}) \geq e^{-\zeta^{\mathrm{c}}(1+d/m)d/m}/(1 + d + d^2/m)$, and substituting for ζ^{c} finishes the proof of theorem 2.

The idea behind the proof of bound for OMN sheds light on the fundamental dilemma that any algorithm with no access to criticality information confronts. On the one hand, because the planner cannot identify critical agents, they all perish. Hence, the planner wishes to make the pool size as small as possible to avoid perishings. On the other hand, the planner wishes to thicken the market so that agents have more matches. In balancing these two opposing forces, we prove that the planner cannot do much to outperform the Greedy algorithm.

These results suggest that criticality information is particularly valuable. This information is necessary to achieve exponentially small loss, and no expansion of information enables an algorithm to perform much better. However, in many settings, agents have privileged insight into their own criticality times. For instance, a rider in a carpooling platform knows the time sensitivity of her request, but the platform does not know that. In such cases, agents may have incentives to misreport whether they are critical, in order to increase their chance of getting matched. The situation is more subtle if agents have waiting costs.

To study this problem, we first formally introduce discounting to our model: an agent receives zero utility if she leaves the market unmatched. If she is matched, she receives a utility of 1 discounted at rate r. More precisely, if $s(a)$ is the sojourn of agent a, then we define the utility of agent a as follows:

$$u(a) \coloneqq \begin{cases} e^{-rs(a)} & \text{if } a \text{ is matched,} \\ 0 & \text{otherwise.} \end{cases}$$

We assume that agents are fully rational and know the underlying parameters and that they believe that the pool is in the stationary distribution when they arrive, but they do not observe the actual realization of the stochastic process. That is, agents observe whether they are critical but do not observe G_t, while the planner observes G_t but does not observe which agents are critical. Consequently, agents' strategies are independent of the realized sample path. Our results are sensitive to this assumption;

for instance, if the agent knew that she had a neighbor or knew that the pool at that moment was very large, she would have an incentive under our mechanism to falsely report that she was critical. This assumption is plausible in many settings; generally, centralized brokers know more about the current state of the market than individual traders. Indeed, frequently agents approach centralized brokers because they do not know who is available to trade with them.

We now exhibit a truthful mechanism without transfers that elicits such information from agents and implements the Patient algorithm.

Definition 4 (Patient mechanism). Ask agents to report when they become critical. When an agent reports being critical, the market maker attempts to match her to a random neighbor. If the agent has no neighbors, the market maker treats her as if she has perished; that is, she will never be matched again.

Each agent a selects a mixed strategy by choosing a function $c_a(\cdot)$; at the interval $[t, t + dt]$ after her arrival, if she is not yet critical, she reports being critical at rate $c_a(t)\,dt$, and when she truly becomes critical she reports immediately. Our main result in this section asserts that if agents are not too impatient, then the Patient mechanism is incentive compatible in the sense that the truthful strategy profile is a strong ϵ-Nash equilibrium.[15]

Theorem 5. Suppose that the market is in the stationary distribution, $d \geq 2$, and $d = \text{polylog}(m)$.[16] If $0 \leq r \leq e^{-d/2}$, then the truthful strategy profile is a strong ϵ-Nash equilibrium for the Patient mechanism, where $\epsilon \to 0$ as $m \to \infty$.

Proof overview.—We sketch the proof here. The full proof can be found in online appendix D. An agent can be matched in one of two ways under the Patient mechanism: either she becomes critical and has a neighbor or one of her neighbors becomes critical and is matched to her. By symmetry, the chance of either happening is the same, because with probability 1 every matched pair consists of one critical agent and one noncritical agent. When an agent declares that she is critical, she is taking her chance that she has a neighbor in the pool right now. By contrast, if she waits, there is some probability that another agent will become critical and be matched to her before she takes her chance of getting matched by reporting to be critical. Consequently, for small r, agents will opt to wait.

There is a hidden obstacle here. Even if one assumes that the market is in a stationary distribution at the point an agent enters, the agent's beliefs about the graph structure and Z_t may change as time passes. In

[15] Any strong ϵ-Nash equilibrium is an ϵ-Nash equilibrium. For a definition of strong ϵ-Nash equilibrium, see definition 5 in online app. D.

[16] The term polylog(m) denotes any polynomial function of $\log(m)$. In particular, d = polylog(m) if d is a constant independent of m.

particular, an agent makes inferences about the current distribution of pool size, conditional on not having been matched yet, and this conditional distribution is different from the stationary distribution. This makes it difficult to compute the payoffs from deviations from truthful reporting. We tackle this problem by using the concentration bounds (see proposition 13 in sec. A2), which limit how much an agent's posterior can be different from her prior. We also focus on strong ϵ-Nash equilibria, which allows small deviations from full optimality.

The key insight of theorem 5 is that remaining in the pool has a "continuation value": the agent, while not yet critical, may be matched to a critical agent. If agents are not too impatient, then the planner can induce truth telling by using punishments that decrease this continuation value. The Patient mechanism sets this continuation value to zero, but in practice softer punishments could achieve the same goal. For instance, if there are multiple potential matches for a critical agent, the planner could break ties in favor of agents who have never misreported. However, such mechanisms can undermine the Erdős-Rényi property that makes the analysis tractable.[17]

VI. Discussion

In this paper, we have studied the problem of dynamic matching in a market with network frictions. The key insight of our analysis is that thickness and information are complements and can be highly valuable. Waiting to thicken the market can yield large gains if the planner can forecast departures accurately. Information about departures is highly valuable if it is feasible to wait. In our simple benchmark, local algorithms perform well relative to algorithms that account for the entire network structure.

One goal of this paper is to identify conditions under which timing concerns are more important than network optimization. To illustrate the limits of these results, we close by discussing two environments in which network optimization is potentially important.

A. Sparse Networks

When evaluating algorithms, our results mainly distinguish between loss that is fractionally small in d and loss that is exponentially small in d. Such bounds have bite when the graph is not extremely sparse ($d < 4$). For instance, when $d = 3$, our bounds imply only that $\mathbf{L}(\mathrm{DAG}) \geq .143$ and

[17] If an agent could be matched even if he misreported previously, then we need to keep track of the edges of that agent off the equilibrium path. However, the fact that the agent was not matched indicates that the agent did not have any edges at the point he misreported, which means that the Markov process of the path does not have a tractable Erdős-Rényi random-graph representation.

$\mathbf{L}(\text{Patient}) \leq .112$, so waiting may not yield large improvements. Similarly, in extremely sparse graphs the upper bound for $\mathbf{L}(\text{Patient})$ leaves room for improvement by considering the network structure. For instance, one could modify the Patient algorithm so that when a critical agent has multiple compatible partners, it breaks ties in favor of partners with lower degree. Simulations indicate that this modified algorithm yields appreciable improvements when d is small (fig. 8, in online app. F).

B. Heterogeneous Types

One limitation of the model is the assumption that agents are ex ante homogeneous. Some of our results crucially depend on this assumption. Suppose instead that agents have types that affect the probability that they are compatible with other agents, as in the following example.

Example 2. There are two types of agents: those who are hard to match (H) and those who are easy to match (E). The probability that two agents of types $i, j \in \{\text{H}, \text{E}\}$ are compatible is p_{ij}, where $p_{\text{HH}} < p_{\text{HE}} = p_{\text{EH}} < p_{\text{EE}}$. Suppose that a critical agent has two neighbors, one H and one E. Then, ceteris paribus, the optimal policy should match the critical agent to the H neighbor.

Clearly, an algorithm that ignored types could have large losses. In this case, a local algorithm would exploit information about neighbors' types but ignore the overall network structure. It is an open question whether local algorithms perform well for heterogeneous types, since our current proofs do not extend straightforwardly to this case.[18]

A recent empirical study of the world's largest carpooling platform (DiDi) by Liu, Wan, and Yang (2019) sheds some light on the role of heterogeneity. Liu, Wan, and Yang (2019) study a two-sided version of our model. Their simulations indicate that for some moderate levels of match heterogeneity, Patient continues to outperform Greedy.[19] However, they also show that for substantial levels of heterogeneity in matches, at least in theory, Greedy can outperform Patient. These findings show that heterogeneity interacts with our results in a subtle way, and a detailed theoretical treatment is an interesting avenue for future research.[20] The techniques we have presented here are a step toward understanding dynamic matching in more general environments.

[18] Even extending our model to two types raises several technical challenges, discussed in the online appendix of Nikzad et al. (2017).

[19] In particular, Liu, Wan, and Yang (2019, 5) write, "[with heterogeneous drivers]. . . . We find that the patient algorithm generates the highest market thickness and achieves the highest match rate at our estimates."

[20] One recent paper by Ashlagi, Nikzad, and Strack (2018) studies a two-type version of our model, similar to the one previously studied in Nikzad et al. (2017), and shows that for the kind of heterogeneity they consider, some of the results presented here will be altered.

Appendix A

Analysis of the Greedy and Patient Algorithms

In the online appendixes, we establish that under both the Patient and Greedy algorithms the random processes Z_t are Markovian, have unique stationary distributions, and mix rapidly to the stationary distribution. These facts are crucial in the analysis of this section. Here, we upper-bound $\mathbf{L}(\text{Greedy})$ and $\mathbf{L}(\text{Patient})$ as functions of d. Note that the result that $\mathbf{L}(\text{Greedy}) \geq 1/(2d+1)$ is simply a corollary of the lower bound on the class of DAG algorithms.

We prove the following theorems.[21]

Theorem 6. For any $\epsilon \geq 0$ and $T > 0$,

$$\mathbf{L}(\text{Greedy}) \leq \frac{\log(2)}{d} + \frac{\tau_{\text{mix}}(\epsilon)}{T} + 6\epsilon + O\left(\frac{\log(m/d)}{\sqrt{dm}}\right), \tag{A1}$$

where $\tau_{\text{mix}}(\epsilon) \leq 2\log(m/d) + \log(2/\epsilon)$.

Theorem 7. For any $\epsilon > 0$ and $T > 0$,

$$\mathbf{L}(\text{Patient}) \leq \max_{z\in[1/2,1]}(z + \tilde{O}(1/\sqrt{m}))e^{-zd} + \frac{\tau_{\text{mix}}(\epsilon)}{T} + \frac{\epsilon m}{d^2} + 2/m, \tag{A2}$$

where $\tau_{\text{mix}}(\epsilon) \leq 8\log(m)\log(4/\epsilon)$.

Setting ϵ small enough that $\epsilon m \to 0$ (e.g., $\epsilon = 1/m^2$ or $\epsilon = 2^{-m}$) implies the second part of theorem 1. This is because when m grows and for $\epsilon m \to 0$, the loss of the Patient algorithm is upper-bounded by $\max_{z\in[1/2,1]} ze^{-zd}$, which is maximized at $z = 1/2$ for $d \geq 2$. Hence, $\mathbf{L}(\text{Patient}) \leq e^{-d/2}/2$. We prove theorem 6 in section A1 and theorem 7 in section A2. Note that the limit results of section III are derived by taking limits from equations (A1) and (A2) (as $T, m \to \infty$).

A1. Loss of the Greedy Algorithm

In this section, we upper-bound $\mathbf{L}(\text{Greedy})$. We crucially exploit the fact that Z_t is a Markov chain and has a unique stationary distribution, $\pi : \mathbb{N} \to \mathbb{R}_+$. Our proof proceeds in three steps. First, we show that $\mathbf{L}(\text{Greedy})$ is bounded by a function of the expected pool size. Second, we show that the stationary distribution is highly concentrated around some point k^*, which we characterize. Third, we show that k^* is close to the expected pool size.

Let $\zeta := \mathbb{E}_{Z\sim\mu}[Z]$ be the expected size of the pool under the stationary distribution of the Markov chain on Z_t. First, observe that if the Markov chain on Z_t is mixed, then the agents perish at the rate of ζ, as the pool is almost always an empty graph under the Greedy algorithm. Roughly speaking, if we run the Greedy algorithm for a sufficiently long time, then the Markov chain on size of the pool mixes,

[21] We use the operators O and $\tilde{O}$ in the standard way. That is, $f(m) = O(g(m))$ iff there exists a positive real number N and a real number m_0 such that $|f(m)| \leq N|g(m)|$ for all $m \geq m_0$. The operator $\tilde{O}$ is similar but ignores logarithmic factors; i.e., $f(m) = \tilde{O}(g(m))$ iff $f(m) = O(g(m)\log^k g(m))$ for some k.

and we get $\mathbf{L}(\text{Greedy}) \simeq \zeta/m$. This observation is made rigorous in the following lemma. Note that as T and m grow, the first three terms become negligible.

LEMMA 8. For any $\epsilon > 0$ and $T > 0$,

$$\mathbf{L}(\text{Greedy}) \leq \frac{\tau_{\text{mix}}(\epsilon)}{T} + 6\epsilon + \frac{1}{m}2^{-6m} + \frac{\mathbb{E}_{Z\sim\pi}[Z]}{m}.$$

The theorem is proved in online appendix E.1. The proof of the above lemma involves lots of algebra, but the intuition is as follows. The $\mathbb{E}_{Z\sim\pi}[Z]/m$ term is the loss under the stationary distribution. This is equal to $\mathbf{L}(\text{Greedy})$ with two approximations. First, it takes some time for the chain to transit to the stationary distribution. Second, even when the chain mixes, the distribution of the chain is not exactly equal to the stationary distribution. The term $\tau_{\text{mix}}(\epsilon)/T$ provides an upper bound for the loss associated with the first approximation, and the term $6\epsilon + (1/m)2^{-6m}$ provides an upper bound for the loss associated with the second approximation.

Given lemma 8, in the rest of the proof we just need to get an upper bound for $\mathbb{E}_{Z\sim\pi}[Z]$. Unfortunately, we do not have any closed-form expression of the stationary distribution, $\pi(\cdot)$. Instead, we use the balance equations of the Markov chain defined on Z_t to characterize $\pi(\cdot)$ and upper-bound $\mathbb{E}_{Z\sim\pi}[Z]$.

Let us rigorously define the transition probability operator of the Markov chain on Z_t. For any pool size k, the Markov chain transits only to state $k + 1$ or state $k - 1$. It transits to state $k + 1$ if a new agent arrives and the market maker cannot match her (i.e., the new agent does not have any edge to the agents currently in the pool), and it transits to state $k - 1$ if a new agent arrives and is matched or an agent currently in the pool becomes critical. Thus, the transition rates $r_{k\to k+1}$ and $r_{k\to k-1}$ are defined as follows:

$$r_{k\to k+1} := m\left(1 - \frac{d}{m}\right)^k, \tag{A3}$$

$$r_{k\to k-1} := k + m\left[1 - \left(1 - \frac{d}{m}\right)^k\right]. \tag{A4}$$

In the above equations, we use the fact that agents arrive at rate m and perish at rate 1 and that the probability of an acceptable transaction between two agents is d/m.

Let us write the balance equation for the above Markov chain (see eq. [A.3] in online app. A for the full generality). Consider the cut separating the states $0, 1, 2, \ldots, k - 1$ from the rest (see fig. A1 for an illustration). It follows that

$$\pi(k-1)r_{k-1\to k} = \pi(k)r_{k\to k-1}. \tag{A5}$$

Now we are ready to characterize the stationary distribution $\pi(\cdot)$. In the following proposition we show that there is a number $k^* \leq \log(2)m/d$ such that under the stationary distribution, the size of the pool is highly concentrated in an interval of length $O((m/d)^{1/2})$ around k^*.[22]

[22] In this paper, log x refers to the natural log of x.

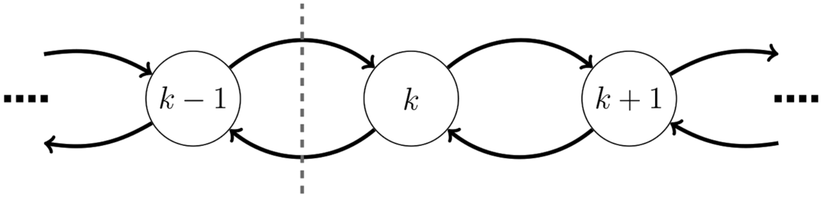

Fig. A1.—Illustration of the transition paths of the Z_t Markov chain under the Greedy algorithm.

Proposition 9. There exists $m/(2d+1) \le k^* < \log(2)m/d$ such that for any $\sigma > 1$,

$$\mathbb{P}_\pi\left[k^* - \sigma\sqrt{2m/d} \;\le\; Z \;\le\; k^* + \sigma\sqrt{2m/d}\right] \ge 1 - O\left(\sqrt{m/d}\right)e^{-\sigma^2}.$$

Proof. Let us define $f : \mathbb{R} \to \mathbb{R}$ as an interpolation of the difference of transition rates over the reals,

$$f(x) \coloneqq m(1-d/m)^x - \{x + m[1-(1-d/m)^x]\}.$$

In particular, observe that $f(k) = r_{k\to k+1} - r_{k\to k-1}$. The above function is a decreasing convex function over nonnegative reals. We define k^* as the unique root of this function. Let $k^*_{\min} \coloneqq m/(2d+1)$ and $k^*_{\max} \coloneqq \log(2)m/d$. We show that $f(k^*_{\min}) \ge 0$ and $f(k^*_{\max}) \le 0$. This shows that $k^*_{\min} \le k^* < k^*_{\max}$.

$$f(k^*_{\min}) \ge -k^*_{\min} - m + 2m(1-d/m)^{k^*_{\min}} \ge 2m\left(1 - \frac{k^*_{\min}d}{m}\right) - k^*_{\min} - m = 0,$$

$$f(k^*_{\max}) \le -k^*_{\max} - m + 2m(1-d/m)^{k^*_{\max}} \le -k^*_{\max} - m + 2me^{-(k^*_{\max})d/m} = -k^*_{\max} \le 0.$$

In the first inequality, we use the Bernoulli inequality, which states that for any $x \le 1$ and any $n \ge 1$, $(1-x)^n \ge 1 - xn$. It remains to show that π is highly concentrated around k^*. In the following lemma, we show that stationary probabilities decrease geometrically.

Lemma 10. For any integer $k \ge k^*$,

$$\frac{\pi(k+1)}{\pi(k)} \le e^{-(k-k^*)d/m},$$

and for any $k \le k^*$, $\pi(k-1)/\pi(k) \le e^{-(k^*-k+1)d/m}$.

This is proved in online appendix E.2.

By repeated application of the above lemma, for any integer $k \ge k^*$, we get[23]

$$\pi(k) \le \frac{\pi(k)}{\pi(\lceil k^* \rceil)} \le \exp\left(-\frac{d}{m}\sum_{i=\lceil k^* \rceil}^{k-1}(i-k^*)\right) \le \exp\left(-d(k-k^*-1)^2/2m\right). \tag{A6}$$

[23] The term $\lceil k^* \rceil$ indicates the smallest integer larger than k^*.

We are almost done. For any $\sigma > 0$,

$$\sum_{k=k^*+1+\sigma\sqrt{2m/d}}^{\infty} \pi(k) \leq \sum_{k=k^*+1+\sigma\sqrt{2m/d}}^{\infty} e^{-d(k-k^*-1)^2/2m} = \sum_{k=0}^{\infty} e^{-d(k+\sigma\sqrt{2m/d})^2/2m}$$

$$\leq \frac{e^{-\sigma^2}}{\min\left\{1/2, \sigma\sqrt{d/2m}\right\}}.$$

The last inequality uses equation (H.1) from online appendix H. We can similarly upper-bound

$$\textstyle\sum_{k=0}^{k^*-\sigma\sqrt{2m/d}} \pi(k).$$

QED

Proposition 9 shows that the probability that the size of the pool falls outside an interval of length $O((m/d)^{1/2})$ around k^* drops exponentially fast as the market size grows.

The following lemma exploits proposition 9 to show that the expected value of the pool size under the stationary distribution is close to k^*.

LEMMA 11. For k^* as in proposition 9,

$$\mathbb{E}_{Z\sim\pi}[Z] \leq k^* + O\left(\sqrt{m/d}\log(m/d)\right).$$

This is proved in online appendix E.3.

Now theorem 6 follows immediately by lemmas 8 and 11, because we have

$$\frac{\mathbb{E}_{Z\sim\pi}[Z]}{m} \leq \frac{1}{m}\left(k^* + O\left(\sqrt{m}\log m\right)\right) \leq \frac{\log(2)}{d} + o(1).$$

A2. *Loss of the Patient Algorithm*

Let $\pi : \mathbb{N} \rightarrow \mathbb{R}_+$ be the unique stationary distribution of the Markov chain on Z_t, and let $\zeta := \mathbb{E}_{Z\sim\pi}[Z]$ be the expected size of the pool under that distribution.

Once more our proof strategy proceeds in three steps. First, we show that $\mathbf{L}$(Patient) is bounded by a function of $\mathbb{E}_{Z\sim\pi}[Z(1 - d/m)^{Z-1}]$. Second, we show that the stationary distribution of Z_t is highly concentrated around some point k^*. Third, we use this concentration result to produce an upper bound for $\mathbb{E}_{Z\sim\pi}[Z(1 - d/m)^{Z-1}]$.

By proposition 6, in online appendix A, at any point in time G_t is an Erdős-Rényi random graph. Thus, once an agent becomes critical, he has no acceptable transactions, with probability $(1 - d/m)^{Z_t-1}$. Since each agent becomes critical at rate 1, if we run Patient for a sufficiently long time, then $\mathbf{L}$(Patient) ≈ $(\zeta/m)(1 - d/m)^{\zeta-1}$. The following lemma makes the above discussion rigorous.

LEMMA 12. For any $\epsilon > 0$ and $T > 0$,

$$\mathbf{L}(\text{Patient}) \leq \frac{1}{m}\mathbb{E}_{Z\sim\pi}\left[Z(1 - d/m)^{Z-1}\right] + \frac{\tau_{\text{mix}}(\epsilon)}{T} + \frac{\epsilon m}{d^2}.$$

This is proved in online appendix E.4.

So in the rest of the proof we just need to lower-bound $\mathbb{E}_{Z\sim\pi}[Z(1-d/m)^{Z-1}]$. As in the Greedy case, we do not have a closed-form expression for the stationary distribution, $\pi(\cdot)$. Instead, we use the balance equations of the Markov chain on Z_t to show that π is highly concentrated around a number k^*, where $k^* \in [m/2, m]$.

Let us start by defining the transition probability operator of the Markov chain on Z_t. For any pool size k, the Markov chain transits only to states $k+1$, $k-1$, or $k-2$. The Markov chain transits to state $k+1$ if a new agent arrives, to state $k-1$ if an agent becomes critical and the planner cannot match him, and to state $k-2$ if an agent becomes critical and the planner matches him.

Remember that agents arrive at rate m, that they become critical at the rate of 1, and that the probability of an acceptable transaction between two agents is d/m. Thus, the transition rates $r_{k\to k+1}$, $r_{k\to k-1}$, and $r_{k\to k-2}$ are defined as follows:

$$r_{k\to k+1} := m, \tag{A7}$$

$$r_{k\to k-1} := k\left(1-\frac{d}{m}\right)^{k-1}, \tag{A8}$$

$$r_{k\to k-2} := k\left[1-\left(1-\frac{d}{m}\right)^{k-1}\right]. \tag{A9}$$

Let us write down the balance equation for the above Markov chain (see eq. [A.3] in online app. A for the full generality). Consider the cut separating the states 0, 1, 2, ...k from the rest (see fig. A2 for an illustration). It follows that

$$\pi(k)r_{k\to k+1} = \pi(k+1)r_{k+1\to k} + \pi(k+1)r_{k+1\to k-1} + \pi(k+2)r_{k+2\to k}. \tag{A10}$$

Now we can characterize $\pi(\cdot)$. We show that under the stationary distribution, the size of the pool is highly concentrated around a number k^*, where $k^* \in [m/2-2, m-1]$. Remember that under the Greedy algorithm, the concentration was around $k^* \in [m/(2d+1), \log(2)m/d]$, whereas here it is at least $m/2$.

Proposition 13 (Patient concentration). There exists a number $m/2-2 \le k^* \le m-1$ such that for any $\sigma \ge 1$,

$$\mathbb{P}_\pi\left[k^* - \sigma\sqrt{4m} \le Z\right] \ge 1 - 2\sqrt{m}e^{-\sigma^2},$$
$$\mathbb{P}\left[Z \le k^* + \sigma\sqrt{4m}\right] \ge 1 - 8\sqrt{m}e^{-\sigma^2\sqrt{m}/(2\sigma+\sqrt{m})}.$$

The proof idea is similar to that for proposition 9, with a little more algebra, and it can be found in online appendix E.5. Since the stationary distribution of Z_t

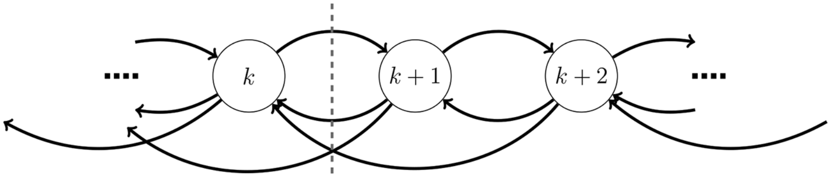

Fig. A2.—Illustration of the transition paths of the Z_t Markov chain under the Patient algorithm.

is highly concentrated around $k^* \in [m/2 - 2, m - 1]$ by the above proposition, we derive the following upper bound for $\mathbb{E}_{Z\sim\pi}[Z(1 - d/m)^Z]$, which is proved in online appendix E.6.

LEMMA 14. For any $d \geq 0$ and sufficiently large m,

$$\mathbb{E}_{Z\sim\pi}\left[Z(1 - d/m)^Z\right] \leq \max_{z\in[m/2,m]} (z + \tilde{O}(\sqrt{m}))(1 - d/m)^z + 2.$$

Now theorem 7 follows immediately by combining lemmas 12 and 14.

Appendix B

Analysis of the OPT and OMN Algorithms

We now provide bounds for optimum solutions by proving the following theorems.

THEOREM 15. If $m > 2d$, then for any $T > 0$,

$$\mathbf{L}(\mathrm{OPT}) \geq \frac{1}{2d + 1 + d^2/m}.$$

THEOREM 16. If $m > 2d$, then for any $T > 0$,

$$\mathbf{L}(\mathrm{OMN}) \geq \frac{e^{-d-d^2/m}}{d + 1 + d^2/m}.$$

It is useful to study the evolution of the system in the case of the inactive algorithm, that is, where the algorithm does nothing and no agents ever get matched. We adopt the notation $\tilde{A}_t$ and $\tilde{Z}_t$ to denote the agents in the pool and the pool size, respectively, in this case. Observe that, by definition, for any matching algorithm and any realization of the process,

$$Z_t \leq \tilde{Z}_t. \tag{B1}$$

Using the above equation, the following proposition shows that, for any matching algorithm, $\mathbb{E}[Z_t] \leq m$.

PROPOSITION 17. For any $t_0 \geq 0$,

$$\mathbb{P}\left[\tilde{Z}_{t_0} = \ell\right] \leq \frac{m^\ell}{\ell!}.$$

Therefore, $\tilde{Z}_t$ is distributed as a Poisson random variable of rate $m(1 - e^{-t_0})$, so

$$\mathbb{E}\left[\tilde{Z}_{t_0}\right] = (1 - e^{-t_0})m.$$

This is proved in online appendix C. Now we are ready to bound losses.

B1. Loss of OPT

In this section, we prove theorem 15.

Let ζ be the expected pool size of the OPT algorithm,

$$\zeta := \mathbb{E}_{t\sim\mathrm{unif}[0,T]}[Z_t].$$

Since OPT does not know A_t^c, each critical agent perishes with probability 1. Therefore,

$$\mathbf{L}(\mathrm{OPT}) = \frac{1}{m \cdot T}\mathbb{E}\left[\int_{t=0}^{T} Z_t \, dt\right] = \frac{\zeta T}{mT} = \zeta/m. \tag{B2}$$

To finish the proof, we need to lower-bound ζ by $m/(2d + 1 + d^2/m)$. We provide an indirect proof by showing a lower bound on $\mathbf{L}(\mathrm{OPT})$, which in turn lower-bounds ζ.

The key idea is to lower-bound the probability that an agent does not have any acceptable transactions throughout her sojourn, and this directly gives a lower bound on $\mathbf{L}(\mathrm{OPT})$, as those agents cannot be matched under any algorithm, so they will all perish, except those who belong to A_T. Since a conservative upper bound for $\mathbb{E}[A_T]$ is m, we then have that the expected number of perished agents is at least $\mathbb{P}[N(a) = \varnothing](mT - m)$, so $\mathbf{L}(\mathrm{OPT}) \geq \mathbb{P}[N(a) = \varnothing](1 - 1/T)$. Since we are mainly stating our results for large values of T, we continue this proof by taking the limit and assuming that $1 - 1/T \simeq 1$ and then discuss how this will change the final result when we include it explicitly.

Fix an agent $a \in A$. Say a enters the market at a time $t_0 \sim \mathrm{unif}[0, T]$ and $s(a) = t$; we can write

$$\mathbb{P}[N(a) = \varnothing] \geq \int_{t=0}^{\infty} \mathbb{P}[s(a) = t] \cdot \mathbb{E}\left[(1 - d/m)^{|A_{t_0}|}\right] \cdot \mathbb{E}\left[(1 - d/m)^{\left|A^n_{t_0, t+t_0}\right|}\right] dt. \tag{B3}$$

To see the above, note that a does not have any acceptable transactions if she does not have any neighbors upon arrival, and none of the new agents that arrive during her sojourn are connected to her. Using Jensen's inequality, we have

$$\begin{aligned}\mathbb{P}[N(a) = \varnothing] &\geq \int_{t=0}^{\infty} e^{-t} \cdot (1 - d/m)^{\mathbb{E}[Z_{t_0}]} \cdot (1 - d/m)^{\mathbb{E}[|A^{\mathrm{n}}_{t_0, t+t_0}|]} \, dt \\ &= \int_{t=0}^{\infty} e^{-t} \cdot (1 - d/m)^{\zeta} \cdot (1 - d/m)^{mt} \, dt.\end{aligned}$$

The last equality follows by the fact that $\mathbb{E}[|A^{\mathrm{n}}_{t_0, t+t_0}|] = mt$. Since $1 - d/m \geq e^{-d/m - d^2/m^2}$ for $d/m < 0.5$,[24] we have

$$\mathbf{L}(\mathrm{OPT}) \geq \mathbb{P}[N(a) = \varnothing] \geq e^{-\zeta(d/m + d^2/m^2)} \int_{t=0}^{\infty} e^{-t(1+d+d^2/m)} \, dt \geq \frac{1 - \zeta(1 + d/m)d/m}{1 + d + d^2/m}. \tag{B4}$$

Putting equations (B2) and (B4) together, for $\beta \coloneqq \zeta d/m$ we get

$$\mathbf{L}(\mathrm{OPT}) \geq \max\left\{\frac{1 - \beta(1 + d/m)}{1 + d + d^2/m}, \frac{\beta}{d}\right\} \geq \frac{1}{2d + 1 + 2d^2/m},$$

[24] This is because $1 - x = e^{-x-x^2}$ has a solution at $x \simeq 0.68$ and $1 - x > e^{-x-x^2}$ for smaller values of x.

where the second inequality follows by letting $\beta = d/(2d + 1 + 2d^2/m)$ be the minimizer of the middle expression.[25]

B2. *Loss of OMN*

In this section, we prove theorem 16.

The proof is very similar to that of theorem 15. Let ζ be the expected pool size of the OMN,

$$\zeta := \mathbb{E}_{t\sim \mathrm{unif}[0,T]}[Z_t].$$

By condition (B1) and proposition 17,

$$\zeta \leq \mathbb{E}_{t\sim \mathrm{unif}[0,T]}\left[\tilde{Z}_t\right] \leq m.$$

Now fix an agent $a \in A$, and let us lower-bound the probability that $N(a) = \varnothing$. Say a enters the market at time $t_0 \sim \mathrm{unif}[0, T]$ and $s(a) = t$; then,

$$\begin{aligned}\mathbb{P}[N(a) = \varnothing] &= \int_{t=0}^{\infty} \mathbb{P}[s(a) = t] \cdot \mathbb{E}\left[(1 - d/m)^{Z_{t_0}}\right] \cdot \mathbb{E}\left[(1 - d/m)^{\left|A^n_{t_0,t+t_0}\right|}\right] dt \\ &\geq \int_{t=0}^{\infty} e^{-t}(1 - d/m)^{\zeta + mT}\, dt \geq \frac{e^{-\zeta(1+d/m)d/m}}{1 + d + d^2/m} \geq \frac{e^{-d-d^2/m}}{1 + d + d^2/m},\end{aligned}$$

where the first inequality uses Jensen's inequality and the second inequality uses the fact that when $d/m < 0.5$, $1 - d/m \geq e^{-d/m-d^2/m^2}$.

References

Akbarpour, Mohammad, and Shengwu Li. 2019. "Smoothed Clearing Algorithms in Dynamic Matching Markets." Manuscript.

Akkina, Sanjeev K., Heather Muster, Eugenia Steffens, S. Joseph Kim, Bertram L. Kasiske, and Ajay K. Israni. 2011. "Donor Exchange Programs in Kidney Transplantation: Rationale and Operational Details from the North Central Donor Exchange Cooperative." *American J. Kidney Diseases* 57 (1): 152–58.

Anderson, Ross, Itai Ashlagi, David Gamarnik, and Yash Kanoria. 2015. "A Dynamic Model of Barter Exchange." In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, edited by Piotr Indyk, 1925–33. Philadelphia: Soc. Indus. and Appl. Math.

APKD (Alliance for Paired Kidney Donation). 2017. "Join Transplant Center Network." http://paireddonation.org/healthcare-professionals/healthcare-professionals-process/.

Arnosti, Nick, Ramesh Johari, and Yash Kanoria. 2014. "Managing Congestion in Decentralized Matching Markets." In *EC '14: Proceedings of the Fifteenth ACM Conference on Economics and Computation*, 451. New York: Assoc. Computing Machinery.

[25] If we do not drop the $1 - 1/T$ term, then the only change would be that the first term inside the max function has an additional $1 - 1/T$ term. This changes the final bound to $\mathbf{L}(\mathrm{OPT}) \geq 1/(\{1 + [T/(T - 1)]\}d + 1 + \{1 + [T/(T - 1)]\}d^2/m)$, which goes to $1/(2d + 1 + 2d^2/m)$ as T grows.

Ashlagi, Itai, Maximilien Burq, Patrick Jaillet, and Amin Saberi. 2018. “Maximizing Efficiency in Dynamic Matching Markets.” Manuscript. https://arxiv.org/abs/1803.01285.

Ashlagi, Itai, Patrick Jaillet, and Vahideh H. Manshadi. 2013. “Kidney Exchange in Dynamic Sparse Heterogenous Pools.” In *EC ’13: Proceedings of the Fourteenth ACM Conference on Electronic Commerce*, 25–26. New York: Assoc. Computing Machinery.

Ashlagi, Itai, Afshin Nikzad, and Philipp Strack. 2018. “Matching in Dynamic Imbalanced Markets.” Working paper, Cornell Univ. https://arxiv.org/abs/1809.06824.

Awasthi, Pranjal, and Tuomas Sandholm. 2009. “Online Stochastic Optimization in the Large: Application to Kidney Exchange.” In *IJCAI’09: Proceedings of the 21st International Joint Conference on Artifical Intelligence*, edited by Hiroaki Kitano, 405–11. San Francisco: Morgan Kaufmann.

Baccara, Mariagiovanna, SangMok Lee, and Leeat Yariv. 2015. “Optimal Dynamic Matching.” Working paper. http://dx.doi.org/10.2139/ssrn.2641670.

Bloch, Francis, and Nicolas Houy. 2012. “Optimal Assignment of Durable Objects to Successive Agents.” *Econ. Theory* 51 (1): 13–33.

Blum, Avrim, John P. Dickerson, Nika Haghtalab, Ariel D. Procaccia, Tuomas Sandholm, and Ankit Sharma. 2015. “Ignorance Is Almost Bliss: Near-Optimal Stochastic Matching with Few Queries.” In *EC ’15: Proceedings of the Sixteenth ACM Conference on Economics and Computation*, 325–42. New York: Assoc. Computing Machinery.

Budish, Eric, Peter Cramton, and John Shim. 2015. “The High-Frequency Trading Arms Race: Frequent Batch Auctions as a Market Design Response.” *Q.J.E.* 130 (4): 1547–621.

Das, Sanmay, John P. Dickerson, Zhuoshu Li, and Tuomas Sandholm. 2015. “Competing Dynamic Matching Markets.” In *Proceedings of the Third Conference on Auctions, Market Mechanisms and Their Applications (AMMA)*, edited by Scott Duke Kominers and Lirong Xia, 2–12. Trent: European Alliance for Innovation.

Dickerson, John P., Ariel D. Procaccia, and Tuomas Sandholm. 2012. “Dynamic Matching via Weighted Myopia with Application to Kidney Exchange.” In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 1340–46. Palo Alto, CA: Assoc. Advancement Artificial Intelligence.

Doval, Laura. 2016. “A Theory of Stability in Dynamic Matching Markets.” Working paper.

Du, Songzi, and Yair Livne. 2014. “Rigidity of Transfers and Unraveling in Matching Markets.” Manuscript.

Ebrahimy, Ehsan, and Robert Shimer. 2010. “Stock-Flow Matching.” *J. Econ. Theory* 145 (4): 1325–53.

Erdős, Paul, and Alfréd Rényi. 1960. “On the Evolution of Random Graphs.” *Bull. Inst. Internat. Statis.* 38 (4): 343–47.

Goel, Gagan, and Aranyak Mehta. 2008. “Online Budgeted Matching in Random Input Models with Applications to Adwords.” In *SODA ’08: Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, 982–91. Philadelphia: Soc. Indus. and Appl. Math.

Kadam, Sangram V., and Maciej H. Kotowski. 2018. “Multiperiod Matching.” *Internat. Econ. Rev.* 59 (4): 1927–47.

Karp, Richard M., Umesh V. Vazirani, and Vijay V. Vazirani. 1990. “An Optimal Algorithm for On-Line Bipartite Matching.” In *STOC ’90: Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing*, 352–58. New York: Assoc. Computing Machinery.

Kurino, Morimitsu. 2009. "House Allocation with Overlapping Agents: A Dynamic Mechanism Design Approach." Econ. Res. Paper no. 2009,075, Univ. Jena.

Leshno, Jacob D. 2012. "Dynamic Matching in Overloaded Systems." Working paper.

Liu, Tracy, Zhixi Wan, and Chenyu Yang. 2019. "The Efficiency of a Dynamic Decentralized Two-Sided Matching Market." Working paper. http://dx.doi.org/10.2139/ssrn.3339394.

Loertscher, Simon, Ellen V. Muir, and Peter G. Taylor. 2016. "Optimal Market Thickness and Clearing." Working paper.

Manshadi, Vahideh H., Shayan Oveis Gharan, and Amin Saberi. 2012. "Online Stochastic Matching: Online Actions Based on Offline Statistics." *Math. Operations Res.* 37 (4): 559–73.

Mehta, Aranyak, Amin Saberi, Umesh Vazirani, and Vijay Vazirani. 2007. "Adwords and Generalized Online Matching." *J. ACM* 54 (5): 22. https://doi.org/10.1145/1284320.1284321.

Nikzad, Afshin, Mohammad Akbarpour, Michael A. Rees, and Alvin E. Roth. 2017. "Financing Transplants' Costs of the Poor: A Dynamic Model of Global Kidney Exchange." Working paper, Stanford Univ.

Özkan, Erhun, and Amy R. Ward. 2017. "Dynamic Matching for Real-Time Ridesharing." Working paper.

Parkes, David C. 2007. "Online Mechanisms." In *Algorithmic Game Theory*, edited by Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V. Vazirani, 411–39. Cambridge: Cambridge Univ. Press.

Parkes, David C., and Satinder Singh. 2003. "An MDP-Based Approach to Online Mechanism Design." In *Advances in Neural Information Processing Systems 16: Proceedings of the 2003 Conference*, edited by Sebastian Thrun, Lawrence K. Saul, and Bernhard Schölkopf, 791–798. Cambridge, MA: MIT Press.

Roth, Alvin E., Tayfun Sönmez, and M. Utku Ünver. 2004. "Kidney Exchange." *Q.J.E.* 119 (2): 457–88.

———. 2005. "Pairwise Kidney Exchange." *J. Econ. Theory* 125 (2): 151–88.

Shimer, Robert, and Lones Smith. 2001. "Matching, Search, and Heterogeneity." *Advances Macroeconomics* 1 (1): 5. https://doi.org/10.2202/1534-6013.1010.

Sönmez, Tayfun, and M. Utku Ünver. 2015. "Enhancing the Efficiency of and Equity in Transplant Organ Allocation via Incentivized Exchange." Working paper. http://dx.doi.org/10.2139/ssrn.2551344.

Su, Xuanming, and Stefanos A. Zenios. 2005. "Patient Choice in Kidney Allocation: A Sequential Stochastic Assignment Model." *Operations Res.* 53 (3): 443–55.

UNOS (United Network for Organ Sharing). 2015. "Kidney Paired Donation Pilot Program: Five Years of Lifesaving Service." https://www.unos.org/kidney-paired-donation-pilot-program-five-years-of-lifesaving-service/.

Ünver, M. Utku. 2010. "Dynamic Kidney Exchange." *Rev. Econ. Studies* 77 (1): 372–414.

Yan, Chiwei, Helin Zhu, Korolko, Nikita, and Dawn Woodard. 2019. "Dynamic Pricing and Matching in Ride-Hailing Platforms." *Naval Res. Logistics.* https://doi.org/10.1002/nav.21872.

Zenios, Stefanos A. 2002. "Optimal Control of a Paired-Kidney Exchange Program." *Management Sci.* 48 (3): 328–42.

Zhou, Haijun, and Zhong-can Ou-Yang. 2003. "Maximum Matching on Random Graphs." https://arxiv.org/pdf/cond-mat/0309348.pdf.