Dynamic Matching Algorithms

by

Maximilien Burq

M.S., Ecole Polytechnique (2015)

Submitted to the Sloan School of Management
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Operations Research

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2019

Author . . . .
Sloan School of Management
September 17, 2018

Certified by . . . .
Itai Ashlagi
Assistant Professor of Management Science and Engineering
Stanford University
Thesis Supervisor

Certified by . . . .
Patrick Jaillet
Dugald C. Jackson Professor of Electrical Engineering and Computer Science
Co-director, Operations Research Center
Thesis Supervisor

Accepted by . . . .
Dimitris Bertsimas
Boeing Professor of Operations Research
Co-director, Operations Research Center

Dynamic Matching Algorithms

by

Maximilien Burq

Submitted to the Sloan School of Management
on September 17, 2018, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Operations Research

Abstract

We study marketplaces in which participants arrive over time, looking to interact with each other. While such interactions have historically been decentralized, the past few years have seen a dramatic increase in the number of internet-enabled platforms which facilitate the process of connecting together, or matching, sets of two or more participants. We will focus mainly on *centralized* matching markets such as *kidney exchange* and *carpooling* platforms. In such platforms, the algorithm which determines whom to match and when to do so plays an important role in the efficiency of the marketplace.

In the first part, we study the interface between the participant heterogeneity, the types of matchings that are allowed, and the frequency at which the platform computes the allocations. We provide an empirical analysis of the effect of match frequency based on data from major US Kidney exchange programs. We then study models that enable us to compare the participants' match rates and waiting times under varying matching policies. We show both in theory and in practice that matching quickly can be beneficial, compared to policies which try to increase opportunities for optimization through artificial waiting.

Until now, the theory of matching algorithms has focused mostly on static environments and little is known in the case where all participants arrive and depart dynamically. In our second part, we help bridge this gap by introducing a new theoretical problem for dynamic matching when anyone can arrive online. We provide new algorithms with state-of-the-art theoretical guarantees, both in the case of adversarial and random order inputs. Finally, we show that these algorithms perform well on kidney exchange and carpooling data.

Thesis Supervisor: Itai Ashlagi
Title: Assistant Professor of Management Science and Engineering
Stanford University

Thesis Supervisor: Patrick Jaillet
Title: Dugald C. Jackson Professor of Electrical Engineering and Computer Science
Co-director, Operations Research Center

Acknowledgments

Throughout my Ph.D. in the Operations Research Center at MIT, I am extremely fortunate to have been co-advised by Itai Ashlagi and Patrick Jaillet. I would like to express my deepest thanks for their unwavering guidance and support during these past four years. Patrick, thank you for pushing me to work on the hard problems, and for believing in my success even when I didn't. Itai, thank you for the countless hours spent together exploring new research directions, for always being so enthusiastic, and for advising me to work on the problems that matter most. Thank you both for being not only incredibly inspiring academic advisors but also caring mentors.

I would also like to thank my other two outstanding collaborators, Vahideh Manshadi and Amin Saberi. To Vahideh, thank you for all the hours working on the whiteboard together, for guiding me through the research process, and for keeping my hopes and motivation up. To Amin, thank you for the time we spent thinking together about the new problems in this thesis, and for teaching me so much on approximation and online algorithms.

There are many other faculty who have had a significant impact on my time and research at the ORC. Thank you in particular to David Gamarnik and Asu Ozdaglar for serving on my thesis and general exam committee. I also had the pleasure to work with many other incredible people over the course of two summer internships. I would like to thank Martin Ryssen, Martin Prillard, Ricky Chachra, Chinmoy Dutta and Daniel Freund for everything I learned with them and for our collaborations.

My time at MIT would not have been the same without all the friends I met during my time there. I want to thank the crew of fellow X-(wo)men with whom I embarked on this journey: Oscar Girardot, Emmanuel Sieffert, Cédric Bozonnat, Lucie Reymondet, Sébastien *Riri* Martin, Sébastien *Fifi* Dubois and Sébastien *Loulou* Boyer. To Manu, thank you for your enthusiasm, for the pool games, and for making Tang a good experience. To Seb and Lucie, thank you for a great year in Otis and the best toga party Boston has ever seen. To my Versailles roommates, Spencer Wyant, Max Biggs and Hugh Carson, thank you for the countless late night discussions and for teaching me about GCs, sailing and right whales. To

Sébastien Martin, thank you for encouraging me to pursue the Ph.D. in the first place, for all our conversations, the life advice, and most importantly for our friendship. A very big thanks goes to all my ORC friends, and in particular: Andrew Andy Li, Anna Papush, Charles Thraves, Daniel Smilkov, Alex Sahyoun, Mapi Testa, Joey Huchette, Stefano Traca, Zach Owen, Virgile Galle, Colin Pawlowski, Ludovica Rizzo, Arthur Flajolet, Pierre Bertrand, Ali Aouad, Mathieu Dahan, Emily Meigs and Elisabeth Paulson. Thank you for the trips, the halloween group costumes, Colin's birthday, and for giving me a second family on this side of the Atlantic. A special thanks goes to my squash partners, Virgile Galle, Victor Prost, Manu Sieffert, Sébastien Boyer and Antoine Dedieu - can't wait to beat you next time we play! To all my other friends, old and new, Vincent Bogart, Laurent Cornaggia, Wanyi Li, Jacques de Chalendar, Paige Beaton, Pieter Coulier, Jean-Claude Angles and Yann Giret, thank you for the good times we had together - looking forward to many more.

I want to acknowledge all that my family has done and continues to do for me. To my parents, Anne-Hélène and Nicolas, thank you for giving me both the opportunity and the will to study and work in areas that will make a positive difference. To my brothers François and Adrien, thank you for all the heated discussions, the laughs and all the adventures we have together.

Finally, my biggest thanks go to Cécile Casses. Thank you for always being there for me, for your friendship and your love. This thesis is dedicated to you.

Thank you.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Algorithms for dynamic matching

We study marketplaces in which participants arrive over time, looking to interact with each other, and may leave if they are not able to find a suitable partner after some time. While such interactions have historically been decentralized, the past few years have seen a dramatic increase in the number of internet-enabled platforms which facilitate the process of connecting, or matching, sets of two or more participants. These platforms typically arise whenever transactions cannot simply be performed using monetary transfers. Several reasons may contribute to this:

- Cultural reasons: Dating for instance does not usually involve monetary transactions between participants. Other examples can include finding an opponent in online games, book exchanges, etc.

- Ethical or regulatory reasons: Typically, it is illegal in most countries around the world to buy and sell kidneys[1]. Similarly, one cannot allocate affordable housing to the highest bidder.

- Technical constraints: For example in ride-sharing or carpooling platforms, the speed at which matches happen often requires specialized algorithms.

In this thesis, we will focus on the *centralized* setting where the final matching decision is taken by the platform.[2] The platform's matching problem is made difficult by three key characteristics: decisions have to be made dynamically, participants are heterogeneous, and there is uncertainty regarding future arrivals and departures.

Dynamic Decisions: We are concerned with instances where participation in the platform is highly dynamic: participants arrive and leave over time, and they may only be offered matches while they are present. The platform thus needs to make decisions dynamically.

[1] To the best of our knowledge, only the Islamic Republic of Iran has a legal market for kidneys.

[2] In some other important cases, the planner is only able to offer suggestions to participants who ultimately decide on whom to match. This creates a new set of challenges such as accounting for participant preferences and strategic decisions.

Participant Heterogeneity: The value of matching two participants depends on both parties' characteristics and preferences. Furthermore, some participants may find it more difficult than others to find a suitable match. They may be also be more at risk than others of departing soon after arriving.

Future Uncertainty: The dynamic decisions are made difficult by a high degree of uncertainty regarding future arrivals and departures. The goal is to tradeoff the benefits of matching two participants now against waiting for a hypothetical better match option.

Several marketplaces face such a problem. Real time ride-hailing platforms have to decide which driver to dispatch for a given passenger request. Such platforms may also offer the option to carpool passengers and hence match passengers with each other, to be transported in the same car. Kidney exchange platforms face the problem of matching patient-donor pairs in a way that allows patients to receive a kidney transplant. The common challenge in all these applications comes from the uncertainty associated with the value of waiting to match later, as this depends on unknown future participant arrivals.

1.2 Motivation

Throughout this thesis, we will be concerned with the theory of dynamic matching algorithms as well as the implications for real-world platforms that rely on these algorithms. In this section, we outline a few of these platforms, with an emphasis on *Kidney Exchange* and *Carpooling*.

1.2.1 Kidney exchange

Worldwide, over 2 million people received some form of renal replacement therapy in 2010, and this likely represents only 10% of those who need it [38, 52]. An estimated 1 million die annually from untreated kidney failure [91]. [3]

[3] In the US, as of 2017, more than 740,000 people suffer from End Stage Renal Disease (ESRD), also known as kidney failure [127], and more than 100,000 patients are on the waiting list for cadaver kidneys [125].

The preferred treatment for kidney failure is transplantation [123]. However there is an acute shortage of kidneys for transplants[4] and many live donors are incompatible with their intended recipients.[5] Kidney exchange allows such patient-donor pairs to swap donors so that each patient can receive a kidney from a compatible donor. There have been recent efforts to create large platforms to increase opportunities for these exchanges [93, 103].

Two or more incompatible pairs can form a cyclic exchange so that each patient can receive a kidney from a compatible donor. In addition, a chain of exchanges can be initiated by a non-directed donor (an altruistic donor who does not designate a particular intended patient).[6] Because monetary transfers are not allowed, it is also impossible to enforce contracts that may arise in kidney exchange. Therefore transplants that occur as a result of an exchange need to be conducted simultaneously in order to avoid the situation where a donor decides not to proceed with the donation after her intended patient has received a kidney.[7] Logistical constraints in turn limit possible exchanges to cycles involving at most 3-4, which correspond to 6-8 simultaneous surgeries [103, 104, 105]. Chains, on the other hand, can be done non-simultaneously, as a participant's decision not to donate to the next pair in the chain does not make said pair worse off, and therefore do not need to be bounded in length [100, 106].

1.2.2 Carpooling

Traffic congestion is a severe problem in metropolitan areas around the world. A resident in Los Angeles is estimated to lose around $6,000 per year due to spending extra hours in traffic (Economist 2014). This does not account for extra carbon emissions. Two possible ways to relieve congestion are pricing [129] and carpooling; online platforms and other technological

[4] In the US, as of 2017, the average waiting time to receive a kidney from a deceased donor is 3 to 5 years. Every year, roughly 35,000 people are added to the waiting list, while only 16,000 receive a kidney.

[5] For a transplant to take place, the donor and intended recipient need to be both blood-type and tissue-type compatible (i.e. the patient has no antibodies to the donor's antigenes).

[6] See [119] for a detailed description of kidney exchange.

[7] To avoid confusion, we will use the pronouns *she, her* instead of the plural *they, their* whenever referring to a person of undetermined gender.

advances are now available to assist with these tasks [94].[8]

Online platforms now offer the option to share rides. An immediate benefit is that passengers who share rides pay a lower price for the trip. However, the passenger may also experience a disutility from additional waiting, detours, and less privacy. Facing these tradeoffs, ride-sharing platforms and carpooling applications seek to increase the volume of ride-sharing, which will in turn help in reducing congestion. This requires sophisticated methods that perform the real-time match decision in addition to other routing and pricing problems.

1.2.3 *Centralized* matching

While most of the models and numerical experiments in this thesis will relate to either kidney exchange or carpooling, some of the insights and algorithms developed here may also apply in some other settings.

The problem of dynamically scheduling jobs that arrive sequentially on a finite number of resources has numerous applications, ranging from physical to computational tasks (internet routers, cloud computing). Similarly, the problem of allocating internet advertisements while a webpage is being loaded can be viewed as a highly dynamic matching problem. Finally, platforms that offer online games often need to match participants to their opponents and teammates. This requires trading off the quality of the match (i.e. matching opponents of similar levels, or teams with complementary skills) with a low waiting time.

1.2.4 *Decentralized* matching

There are many other instances where the platform may only offer suggestions or recommendations to the users, and the matching decision is conducted in a decentralized manner. This often happens in the form of a recommendation system or a search algorithm. A few examples include: e-retail, on demand video services, car or house sharing platforms, barter

[8] Ostrovsky and Schwarz [94] discuss the complementarities between autonomous vehicles, carpooling and pricing.

exchange platforms, affordable housing allocation, cadaver organ allocation as well as a few decentralized carpooling platforms.

While this decentralized setting is not the focus of this thesis, we hope that some of the insights which we obtain in the centralized setting will be applicable there as well.

1.3 Mathematical framework

In this section we present a mathematical framework that encapsulates all the dynamic matching problems we will study. This enables us to abstract away key aspects of the dynamic matching problem from some of the platform specificities.

1.3.1 General framework

Compatibility Graph. We model the participants in the marketplace as vertices in a *compatibility graph* G. Depending on the application, G may be a directed or undirected graph.[9] Each vertex $i \in \mathcal{V}$ arrives over time and is associated with an arrival time t_i and a departure time d_i. Vertices reveal the edges that they are adjacent to upon arrival.[10] We denote $\mathcal{E} \subset \mathcal{V}^2$ the set of edges in G. For each pair $i, j \in \mathcal{E}$, we denote the edge from i to j by (i, j), and associate the value $v_{i,j}$.[11]

The compatibility graphs allowed will differ depending on the model. In Part I, we focus on graphs that are unweighted: $v_{i,j} \in \{0, 1\}$, whereas in Part II, any non-negative edge weight is allowed. The choice of how graphs are generated will also vary: in Chapters 3 and 8 we will use real-world datasets to generate the compatibility graphs. In Chapters 4 and 5, we will assume a stochastic Erdős-Renyi model for G. In Chapters 6, and 7 we will design algorithms that perform well over any graph G.[12]

[9] Without loss of generality, one could assume that the graph is always directed, and sometimes add the additional constraint that each forward arc $i \to j$ comes with the associated backward arc $j \to i$.

[10] Throughout this thesis, the word *edge* will be used both in the case of a directed or undirected graph. It will be clear from the context which one is being used.

[11] When G is undirected, our convention will be that $v_{i,j} = v_{j,i}$.

[12] This is sometimes called the *adversarial* setting, as the algorithm needs to perform well even when the input graph is chosen in the worst possible way.

Allowed Matchings. A *matching rule* R determines whether a set of edges $m \subset \mathcal{E}$ constitutes a *matching*. In practice, different types of platforms will allow different kinds of matchings. For example, if we wish to match kidney patient-donor pairs, a matching can be a directed cycle of length less than 4. If we wish to match passengers to drivers, a matching can be an edge from a driver to a passenger that satisfies some geographic constraints.

In Part I, we focus on a directed compatibility graph G. The matching rule will be to accept short cycles and/or chains. Cycles have to have a length of at most 3 of the form: $\{(i,j),(j,i)\}$, or $\{(i,j),(j,k),(k,i)\}$. Chains can be of arbitrary lengths, of the form $\{(i_0,i_1),(i_1,i_2),...,(i_l,i_{l+1})\}$ where l is the length of the chain and i_0 is constrained to belong to a subset of *bridge* vertices.

In Part II, we will focus on an undirected compatibility graph. The matching rule will be to accept any single edge $\{(i,j)\}$. Note that the switch to an undirected graph is mainly for the ease of exposition, we could just as well consider a directed graph with a matching rule that accepts cycles of length two $\{(i,j),(j,i)\}$.

We will say that a vertex is in a matching s if it is adjacent to at least one edge in s. A valid *allocation* is a set of valid *matchings* $a = \{m_1, ..., m_k\}$ where no vertex is included in more than one matching m_i. In other words, it is not possible to include a single vertex in more than one cycle or chain. We will say that a vertex is in an allocation if it is in one of its matchings.

The value of a matching m is the sum of the values of its edges: $v(m) = \sum_{e \in m} v_e$. The value of an allocation a is the sum of the values of its matchings: $v(a) = \sum_{m \in a} v(m)$.

Time and Departure Model. We will model time to be either discrete or continuous depending on the model, see section 1.4 for a more in-depth discussion. At any time t, the platform can decide to *match* an allocation a. In that case, we remove all the vertices of a from G, and the platform collects $v(a)$. If a vertex i is not matched by time d_i, it departs unmatched: i is removed from G and no value is collected by the platform.[13]

[13] In the case of carpooling, this may be because the participant decided to use another mode of transportation, in the case of kidney exchange, this is often because the patient found a kidney through some other means, or became too sick to safely undergo surgery.

Objective Functions. Depending on the setting, we will assume two different kinds of objectives: the platform may either seek to minimize the expected waiting time of vertices in the graph (this is the case in chapter 5), or to maximize the sum of total value of matchings collected (chapter 4, 6, 7, 8), or a combination of both at the same time (chapter 3)

There are two key difficulties for the platform's decision. First is to decide when to search for allocations, and this will be the topic of chapters 3 and 4. Second is to decide which vertices to include in the allocation, and which ones to keep waiting in the hope of a better match in the future. This will be the topic of chapters 6 and 7. In some instances, the platform is also able to influence the choice of the matching rule R, as well as the nature of the graph G. In chapter 5, we study how these parameters impact the performance of the platform.

In Table 1.1 we provide a concise summary of the models studied, and how they each fit in our mathematical framework.

	Chapter	3	4	5	6	7	8
Graph type	adversarial				x		
	random order					x	
	stochastic		x	x			x
	data-driven	x				x	
Edge weights	$\mathbb{R}^+$				x	x	x
	$\{0, 1\}$	x	x	x			
Matching rule:	2-cycles	x	x	x	x	x	x
	3-cycles	x	x				
	chains	x		x			
Time model:	discrete	x	x		x	x	x
	continuous			x			
Departure process	deterministic				x	x	
	stochastic	x	x		x		x
Objective function:	minimize waiting	x		x			
	maximize value	x	x		x	x	x

Table 1.1: Summary of model characteristics for each chapter.

1.3.2 Notations

Throughout this thesis, in order to be consistent with a number of different streams of literature, we will use the terms *participant*, *agent*, *node* and *vertex* interchangeably to denote a person to be matched. In some settings that relate to kidney exchange, we will also use *patient-donor pair* or simply *pair* to denote a single vertex in our compatibility graph.

Similarly, we will use the terms *market*, *pool* and *compatibility graph* to denote the ensemble of participants who wish to be matched.

Competitive ratios

We define an *instance* I of a dynamic matching problem as a collection of a *matching rule* R, a compatibility graph $G(\mathcal{V}, \mathcal{E})$, and sets of arrival and departure times $(t_i, d_i)_{i \in \mathcal{V}}$. See section 1.3 for more details.

For any instance I and matching algorithm $\mathcal{A}$, we will denote $\mathcal{A}(I)$ to be the value collected on that instance. We will denote by $\mathcal{O}$ the offline algorithm which has information on the whole graph G and the arrival and departure sequence.

In Part II, we will measure the performance of our algorithms using the *competitive ratio.*

Definition 1 (Competitive Ratio)**.** *Fix a class $\mathcal{I}$ of instances. A (possibly randomized) algorithm $\mathcal{A}$ is α-competitive in $\mathcal{I}$ if and only if for all instances $I \in \mathcal{I}$, $\mathbb{E}\left[\mathcal{A}(I)\right] \geq \alpha \mathcal{O}(I)$.*

In Chapter 6, we set $\mathcal{I}$ to contain all graphs G with n vertices, and the additional constraint that there exists a time d such that for all $i \in \mathcal{V}$, $d_i = t_i + d$. In Chapters 7 and 8, we relax Definition 1 to be in expectation over classes of instances:

Definition 2 (Stochastic Competitive Ratio)**.** *Fix a set $\mathcal{D}$ of distributions D over instances I. A (possibly randomized) algorithm $\mathcal{A}$ is α-competitive in $\mathcal{D}$ if and only if for all distributions $D \in \mathcal{D}$, $\mathbb{E}_D\left[\mathcal{A}(I)\right] \geq \alpha \mathbb{E}_D\left[\mathcal{O}(I)\right]$.*

In Chapter 7, for each graph G with n vertices, we define a distribution over instances as follows: select uniformly at random a permutation $\sigma \in S_n$ over n vertices. Fix an integer d and for each vertex i, set $t_i = \sigma(i)$ and $d_i = \sigma(i) + d$. Thus for each graph G, we have a uniform distribution D over $n!$ instances. We set $\mathcal{D}$ to be the set of all such distributions for any G.

In Chapter 6, we first define a *type graph* T. For each vertex x in T, we define an arrival rate λ_x and a departure rate θ_x. We denote $\boldsymbol{\lambda}, \boldsymbol{\theta}$ the corresponding vectors. We then generate arrivals i through a Poisson(λ_x) process, and departures through an Exponential(θ_x) process. This generates a distribution D for each type graph T and vectors $\boldsymbol{\lambda}$, $\boldsymbol{\theta}$. We set $\mathcal{D}$ to be the set of all such distributions.

1.4 Outline

This thesis consists of nine chapters, grouped in two main parts. Chapter 2 presents a brief literature review and outlines the main contributions of this work. In Part I, which consists of Chapters 3, 4 and 5, we study the interface between the types of matchings that are allowed [14], the composition of the market and the frequency at which one should search for allocations. In Part II, which consists of Chapters 6, 7 and 8, we focus on the setting where 2-cycles only are allowed and introduce algorithms that perform well over a wide range of possible graphs. Next we give a brief outline of each chapter.

Chapter 2: Literature Review and Main Contributions

This chapter presents existing work and models in the space of dynamic matching. We also outline the key contributions of this thesis. From a theoretical perspective, we contribute new problems and algorithms to the rich *Online Matching* literature. From a practical standpoint, we contribute new insights and algorithms to the medical literature on *Kidney Exchange* as well as the transportation literature on *Carpooling*.

[14] This is motivated by Kidney Exchange, where an important policy decision is the length of cycles to allow, as well as whether to use chains.

Part I: Interface between Matching Technology, Market Heterogeneity and Match Frequency

Chapter 3: The Effect of Match Run Frequency on the Number of Transplants and Waiting Times in Kidney Exchange

This Chapter is based on the paper Ashlagi et al. [17], which is joint work with Itai Ashlagi, Adam Bingaman, Vahideh Manshadi, David Gamarnik, Cathi Murphey, Alvin E. Roth, Marc L. Melcher, and Michael A. Rees, and is published in the *American Journal of Transplantation.*

In Kidney Exchange programs the frequency of match runs, i.e. the choice of how often the platform searches for matches, may have important consequences for patients in need of a transplant. On one hand, infrequent match runs may lead to fewer sub-optimal matching decisions, and thus more transplants. On the other hand, frequent match runs decrease waiting times, and reduce the likelihood that a patient leaves the program without a transplant.[15] We conducted simulations using clinical data from two Kidney Exchange programs in the US to study how the frequency of match-runs impacts the number of transplants and the average waiting times. In this model, valid matchings can be directed cycles of varying length (2, 3, or 4), as well as directed chains initiated by a special type of vertices called non-directed donors.

We simulate the options facing each of the two registries by repeated resampling from their historical pools of patient-donor pairs and non-directed donors, with arrival and departure rates corresponding to the historical data. We find that longer intervals between match-runs do not increase the total number of transplants, and that prioritizing highly sensitized patients is more effective than waiting longer between match-runs for transplanting highly sensitized patients. While we do not find that frequent match-runs result in fewer transplanted pairs, we do find that an increase in arrival rates of new pairs improves both the fraction of transplanted pairs and waiting times.

[15] This can be for instance because the patient became too sick to undergo surgery.

Chapter 4, Cycle Length, Priority Schemes and Matching Frequency.

This chapter is based on the unpublished manuscript Ashlagi et al. [15], which is joint work with Itai Ashlagi, Patrick Jaillet and Vahideh Manshadi.

In this chapter we leverage insights from Chapter 3, as well as statistics derived from a new dataset to create a theoretical model of *Kidney Exchange* that accounts for both patient-donor heterogeneity and the difficulty that some patients have in finding a suitable donor before they need to depart. Importantly, we make sure that our model still remains tractable analytically.

Using this model as well as some new simulation results, we study how both the priority structure, and the length of cycles that are allowed impact the choice of the optimal match frequency. We find that the benefits of small batches have negligible effect on the number of pairs whose donors are compatible with hard-to-match patients. In other words, the in-degree of pairs with hard-to-match patients stays small in the compatibility graph.[16] Subsequently, when only cycles of length 2 are allowed, waiting does not result in a significant number of additional matches. Surprisingly, we find a different result when cycles of length up to 3 are allowed. Although the degree of hard-to-match pairs is still small, it is not always zero. In some cases, they are compatible to receive from another pair but are not able to give back, precluding a 2-cycle. In that case, waiting for a third pair to arrive may increase the odds of closing the loop.

This suggests that in some circumstances, some amount of *targeted* waiting may be beneficial, even when making *everyone* wait is counter-productive.

Chapter 5, On Matching and Thickness in Heterogeneous Dynamic Markets.

This chapter is based on [18], under revision in *Operations Research*, which is joint work with Itai Ashlagi, Patrick Jaillet and Vahideh Manshadi.

[16] Recall from section 1.2.1 that in Kidney Exchange, patient-donor pairs join the program together, and are treated as a unit

In this chapter, we abstract away from the specifics of *Kidney Exchange* and study dynamic matching in a model with no departures. While all agents are potentially compatible with each other, some are hard-to-match and others are easy-to-match. Agents prefer to be matched as soon as possible and matches are formed either bilaterally or indirectly through chains. We adopt an asymptotic approach and compute tight bounds on the limit of waiting time of agents under myopic policies that differ in matching technology and prioritization.

We find that the market composition is a key factor in the desired matching technology and prioritization level. When hard-to-match agents arrive less frequently than easy-to-match ones (i) bilateral matching is almost as efficient as chains, and (ii) assigning priorities to hard-to-match agents improves their waiting times. When hard-to-match agents arrive more frequently, chains are much more efficient than bilateral matching and prioritization has no impact.

We further study the effect of arrival rates on the average waiting time. Somewhat surprisingly, we find that in a heterogeneous market, under bilateral matching, an increase in the arrival rate of hard-to-match agents has a non-monotone effect on waiting times. This is because, under some market compositions, there is an adverse effect of competition. Our comparative statics shed light on the impact of merging markets and attracting altruistic agents (that initiate chains) or easy-to-match agents. This work uncovers fundamental differences between heterogeneous and homogeneous dynamic markets, and potentially helps policy makers to generate insights on the operations of matching markets such as kidney exchange programs.

Part II: Beyond Myopic Policies: Algorithms for Stochastic and Worst Case Graphs

In the next three chapters, we impose fewer constraints on the underlying compatibility graph. We study algorithms that perform well over a wide class of graphs, when matches are constrained to be between two participants only.

Chapter 6, Maximizing Efficiency in Dynamic Matching Markets.

This chapter is based on the manuscript Ashlagi et al. [20], which is joint work with Itai Ashlagi, Patrick Jaillet, Amin Saberi . It was later merged into [19], with Chinmoy Dutta and Chris Sholley and submited to the *Symposium on Discrete Algorithms.*

In this chapter, we consider matching in the classical graph-theoretic sense: the compatibility graph is undirected and two vertices can be matched if and only if there exists an edge between them. We study the problem of matching agents who arrive at a marketplace over time and leave after d time periods. The planner's goal is to maximize the total value over a finite time horizon. We study matching algorithms that perform well over any sequence of arrivals when there is no a priori information about the match values or arrival times.

Our main contribution is a 0.25-competitive algorithm[17]. The algorithm randomly selects a subset of agents who will wait until right before their departure to get matched, and maintains a maximum-weight matching with respect to the other agents. The primal-dual analysis of the algorithm hinges on a careful comparison between the initial dual value associated with an agent when it first arrives, and the final value after d time steps.

We also show that no algorithm is more than 0.5-competitive. We extend the model to the case in which departure times are drawn independently from an exponential distribution, and establish a 0.125-competitive algorithm in this setting.

Finally, we introduce a new algorithm based on the idea of *Re-Optimization.* We show how many of the ideas from the proof of our main result can be adapted to this new algorithm. Although we are not able to formally provide a competitive ratio above 0.25, we give insights as to why it might be possible to do so.

Chapter 7, Online Matching with Deadlines in Random Order.

This chapter is joint work with Itai Ashlagi and Amin Saberi. It was merged into [19] and submitted to the *Symposium on Discrete Algorithms.*

[17] This means that for any graph G and for any sequence of arrivals, our algorithm collects at least 0.25 of the value that is collected by the *offline* algorithm which knows the future.

The theoretical setting for this chapter is similar to that of Chapter 6. Here we also assume that vertices depart after a fixed deadline of d time steps after their arrival. We further assume that the vertices arrive in random order. Our main result is to show that the *batching* algorithm, which periodically searches for the highest-value allocation, is 0.279-competitive. Our proof first reduces the problem to a special graph covering problem. We then show that an lower bound on the competitive ratio of *Batching* can be computed as the limit of a sequence of Linear Programs. Finally, we prove that it is sufficient to explicitely solve a small number of these LPs, and we provide a computer-aided argument to conclude.

We also extend our proof framework to the setting where the online algorithm is allowed to know which vertices are going to arrive l steps ahead of time. We provide a closed-form competitive ratio that depends on the ratio of l/d. In particular we show that if $l \geq d$, then the batching algorithm performs better than *any* online algorithm that does not have access to the look-ahead information.

We conjecture that *Batching* is in fact 1/2-competitive, and we provide a proof sketch as well as some numerical insights and intuition. Finally, we provide numerical simulation results based on datasets of New York City's taxis and the National Kidney Registry. Our simulations compare *Batching* to the *Re-Optimization* algorithm introduced in Chapter 6.

Chapter 8: A Dynamic Matching Framework with Stochastic Information.

In this chapter we consider the setting where the compatibility graph is sampled from a distribution which is known to the matching algorithm. We formalize this using a *type* graph, from which incoming vertices are sampled.

In the first part, we introduce a *Suggested Matching* algorithm inspired by results on the classical *online matching* problem. We prove that *SM* is constant-competitive algorithms for a few relevant special cases of the problem, and provide a competitive ratio which depends on the ratio of arrival to departure rates in the general case.

In the second part, we frame the problem through the lens of dynamic programming. We first show how ideas from approximate dynamic programming can be leveraged to provide useful algorithms. Finally, we show how we can generalize to real-world cases where the number of *types* grows large or even infinite.

Chapter 2

Main Contributions and Literature Overview

This thesis builds on previous work from many fields, such as Computer Science, Economics, Transportation Science, Queueing Theory, etc. The goal of this chapter is not to provide an extensive review, but we will summarize a few results that are relevant to our work, and highlight our contributions.

First, there is a large body of literature on Matching Algorithms, both Static (2.1.1) and Online (2.1.2). Also relevant is the large literature on Queueing (2.1.4) and on Job Scheduling (2.1.3). Finally we will present some of the recent advances in the literature on two applications: Kidney Exchange (2.3) and Carpooling (2.4).

2.1 Matching

2.1.1 Offline matching

A large literature has been dedicated to the static - offline - matching problem with heterogeneous match values. A particular problem is finding a maximum-weight matching problem efficiently. Classic algorithms that have been proposed include the Hungarian [74], Edmond's [51] and auction algorithms [26, 41]. For the case, in which agents have ordinal preferences, Gale and Shapley [60] proposed the Deferred Acceptance algorithm which finds a stable matching. Our work in Chapter 6 builds on these algorithms by maintaining a tentative maximum-weight matching over time, and matches are made final when a seller is about to depart. For reviews of the literature on static matching markets, with participant preferences, see [108, 120].

2.1.2 Online matching

The problem of online matching arises naturally in information technology applications such as online advertising in which advertisements need to be assigned instantly to queries searched or webpages viewed by users. The study of online matching was initiated by Karp et al. [72], in which they analyze the problem in adversarial settings with no probabilistic information about the graph. Follow up work has provided new tools for the analysis of this online matching problem in the adversarial setting [29, 42]. Several follow up papers, studied the problem beyond the worst case settings. Goel and Mehta [61] study the model in which the underlying graph has unknown distribution. Feldman et al. [54] notice that in applications such as online advertising there is information about the graph structure, and they analyze a model where the graph distribution belongs to a certain class. Jaillet and Lu [67], Manshadi et al. [81] study the same problem with a general known distribution. This work has numerous other extensions, for example to stochastic arrivals and in the adwords context [83]. See [82] for a detailed survey.

Several papers consider the problem of dynamic matching in the edge-weighted case. Feldman et al. [55] find that in the classic online bipartite setting, no algorithm achieves a constant approximation. They introduce a *free disposal* assumption, which allows the online algorithm to discard a matched vertex in favor of a new arriving vertex. They find, based on an algorithm by Lehmann et al. [75], that a greedy algorithm that matches a vertex to the highest marginal vertex, is 0.5-competitive. We build on this result for a special classes of bipartite graphs. Zadimoghaddam [132] provides a new analysis with better bounds. This model has also been extended to incorporate participant choice models [79]. In the adversarial setting Ashlagi et al. [16], Emek et al. [53] study the problem of minimizing the sum of distances between matched vertices and the sum of their waiting times. In their model no vertex leaves unmatched and our model does not account for vertices' waiting times. This work is also related to the literature on allocation of re-usable products [47, 110],

Closely related to our model in Chapter 6 is work that was conducted concurrently by Huang et al. [66], which studies a model similar to the one introduced in Chapter 6, but in the case of a graph with $\{0, 1\}$ edges. This allows them to have departure times that are chosen adversarially, as opposed to our deadline model.

This thesis contributes to this literature in two ways. First, we introduce two new models in Chapters 6 and 7, which enable the study of a new class of problems where the compatibility graph G is not bipartite, and all vertices arrive online. This is particularly relevant to applications such as Kidney Exchange and Carpooling. This work also highlights the importance of studying the departure process, in addition to the arrival structure. Second, we provide state-of-the art algorithms that perform well on edge-weighted graphs even without a *free disposal* assumption. We also show that in addition to good performance guarantees in the worst case, some of these algorithms perform well on real datasets.

2.1.3 Job scheduling

Related to our work are some papers on job or packet scheduling. Jobs arrive online to a buffer, and reveal upon arrival the deadline by which they need to be scheduled. The algorithm can schedule at most one job per time and the value of scheduling a job is independent from the time slot. Chin et al. [36] and Hajek [63] provide constant competitive algorithms. Jeż et al. [68], Li et al. [77] study a special case where the packets have *agreeable deadlines*, i.e. they depart in order of arrival. Bar-Noy et al. [25] study a generalization where jobs may take multiple units of time to be completed, and consume fractional amounts of resources.

Our results in Chapters 6 and 7 relate to this literature in three ways. First, the problem of job scheduling can be seen as a special case of our dynamic matching problem, where time slots and jobs arrive online. Our focus on *edge-weighted* graphs makes the problem more difficult, as it would correspond to jobs yielding different values depending on the time they are scheduled at. Second, our model of deadlines where each vertex stays in the pool for the same number of time steps can be seen as a case of *agreeable deadline*. Finally, our heuristic α-DDA algorithm in Chapter 6 builds on the idea of Jeż et al. [68] of giving a multiplicative "bonus" to vertices that are about to expire.

2.1.4 Matching queues

Our model in Chapter 8 is related to the problem of matching multi-class customers to multi-class servers, studied in queueing literature. For example, Caldentey et al. [34] as well as Adan and Weiss [1] study the assymptotic match rates under the *First-come-first-serve* policy. Further work has focused on compatibility-based optimal control problems where vertices do not leave the system unless they are matched. Objectives that have been studied include minimizing waiting costs [33, 62] or maximizing matching rewards [92].

Our work in Chapter 8 introduces a similar optimal control problem where waiting vertices are at risk of departing unmatched.

2.2 Mechanism and market design

A large stream of literature in mechanism design focuses studies matching in decentralized settings where agents' preferences are richer and may lead to strategic behaviour. Important concepts from the "static" matching literature, such as *stability*, can be adapted to a dynamic setting [50, 70]. In such settings with rich preferences, Baccara et al. [22], Fershtman and Pavan [58] find that policies that match without waiting can be inefficient, since some waiting can improve the quality of matches. Our work contributes to this literature through a better understanding of the settings where waiting may be useful in centralized settings.

Mendelson [86] analyzed the behavior of a clearinghouse in a dynamic market with prices in which sellers and buyers arrive over time according to a given stochastic process. Similar to our model in Chapters 4 and 7, he considers a mechanism in which the clearing prices are computed periodically, and he studies the market behavior for different time (period) scales.

Arnosti and Shi [11], Bloch and Cantala [30], Kaplan [71], Leshno [76] study matching mechanisms in the context of affordable housing allocation. They analyze which lottery and wait-list mechanisms lead to efficient allocations. Agarwal et al. [2], Schummer [115], Su and Zenios [122] study similar questions in the context of cadaver organ allocation mechanisms.

Our work differs from this literature mainly by the fact that preferences are assumed to be known to the centralized planner, and we suppose that agents are truthful when revealing these preferences independently from the allocation mechanism.

2.3 Kidney exchange.

Kidney Exchange was initially suggested by Rapaport [99]. Subsequent studies have looked at a variety of questions. Particularly relevant to this thesis, are papers which study the effect of waiting and frequent matching. Also relevant are work which study the role of longer cycles and chains on overall efficiency.

2.3.1 Waiting and match frequency.

Kidney exchange programs periodically perform run optimization algorithms to find cyclic exchanges among incompatible pairs, and chains initiated by a non-directed donor. Intuitively, an important factor that impacts the number of transplants is the timing and frequency at which these match runs are conducted.

A number of stylized models have been proposed over time, and have largely suggested that matching frequently is optimal.[1] Ünver [126] initiated the study of dynamic kidney exchange. Studying a model with different types and a deterministic compatibility structure across types, he finds that matching upon arrival is near optimal, even though some waiting with certain types to facilitate three-way exchanges adds some benefits. Anderson et al. [8] consider a homogeneous model without departures and find that there is little benefit from waiting before matching under either short cycles and chains.[2] Akbarpour et al. [4] consider a homogeneous model with departures and find that the optimality gap of the policy that matches without waiting remains constant as the match probability decreases.

We contribute to this literature in two ways. In Chapter 3, through data-driven simulations which study the impact of match-run frequency, and show that among polices that match periodically (e.g., every week or every day), high matching frequencies perform best. In Chapter 4, we study a model similar to Anderson et al. [8], and show that making a select subset of vertices wait may perform well.

Chapter 5 builds on these findings, and only analyzes myopic policies that search for a match upon each arrival of a new agent.

Overall, as the literature on dynamic matching is developing, a clearer picture is emerging on the impact of waiting. In completely homogenous settings with no knowledge of departure

[1] Non-myopic policies have also been studied, for example Dickerson et al. [44] study forward-looking polices by casting the dynamic matching problem as a high-dimensional dynamic program, and develop a heuristic to overcome the curse of dimensionality.

[2] The waiting-time scales with the same factor with or without waiting before matching.

times, waiting is ineffective Akbarpour et al. [4], Anderson et al. [9]. However, in models where heterogeneity in compatibility and/or preferences are incorporated, and more general forms of matching is allowed (e.g., 3-way cycles in addition to 2-ways) some positive impacts of waiting can become apparent (See also Section 2.2).

2.3.2 Cycle length and never ending chains.

Roth et al. [103] first proposed a way to organize kidney exchange integrating cycles and chains. Logistical constraints required that cycles involve no more than two patient-donor pairs. Subsequent work suggested that a modest expand of infrastructure, that is allowing only slightly larger, 3-and 4-way exchanges would be efficient (Roth et al. [107], Ashlagi and Roth [12]) in large static pools. These studies assume either implicitly or explicitly that no tissue-type incompatibilities exist.

Awasthi and Sandholm [21] use online stochastic optimization methods to computationally study the allocation in kidney exchange using short cycles in a dynamic setting. Dickerson et al. [43] conduct computational simulations in the dynamic settings to understand the benefit of chains. Dickerson et al. [44] study dynamic optimization and propose an algorithm that assigns weights to different matches using future stochastic sampling. These studies use however dense compatibility graphs (they generate graphs according to Saidman et al. [111] which have been shown in Ashlagi et al. [14] to be very dense as opposed to clinical data). Further, [49] use a novel random walk approach to study the effectiveness of chains in a static random graph with hard and easy-to-match agents.

In a related line of work, Zenios [133] studied a dynamic kidney allocation model where both direct (2-way) and indirect kidney exchange are allowed. [3] The objective is to study the tradeoff between the (lower) quality of a deceased donor (that is assumed to be available immediately) and the cost of waiting for a direct exchange. The exchange model studied in this work ignores the tissue-type compatibility.

Anderson et al. [8] consider a homogeneous model without departures and find that

[3] In an indirect exchange, the donor gives a kidney to the highest priority patient in deceased donor waiting list, and her intended patient gets the highest priority in the waiting list.

when the graph is very sparse, allowing longer cycles result in significantly less waiting than restricting to cycles of length two. They find that allowing chains further reduces the waiting time significantly.

Our contributions to this literature are two-fold. First, Chapter 4 highlights how longer cycles may increase the benefit from non-myopic policies. Second, Chapter 5 help quantify the value of chains. In particular we show that the pool composition impacts the performance of short cycles in a significant way.

2.3.3 Failures.

Another line of work in kidney exchange studies matching in the presence of failure. Many of the match offers given by the exchange programs fail to proceed to actual transplant for various logistical and medical reasons. A few recent papers study kidney exchange with failures, [31, 48, 88]. They use a probabilistic model for failure and assume that each compatible match is only viable with a given probability. They design algorithms to maximize the expected number of successful matches in static settings. Further, [48] computationally study this problem in a dynamic setting using dense graph models.

2.3.4 Heterogeneity and pool composition.

Finally, some work has focused on the effects of the market size and composition on efficiency. For instance, merging markets is often sought as a solution to improve efficiency (see e.g. [32, 118]). It has also been reported that kidney exchange programs attempt to attract easy-to-match pairs [12].

Our heterogeneous model in Chapters 4 and 6 contribute to this literature by highlighting some of the tradeoffs that occur, and how they relate to the types of matching that are allowed.

2.4 Ride sharing and carpooling

There is a rich history of using optimization algorithms in the transportation sector. Recently, there has been a surge of interest in models related to ride-sharing. Santi et al. [113] finds that about 80% of rides in Manhattan could be shared by two passengers. Recently, Ostrovsky and Schwarz [94] discuss the complementarities between autonomous vehicles, carpooling and pricing. Many studies focus on problems related to dynamic pricing [35], routing [27, 130], scheduling [37, 130], rebalancing vehicles [23, 24, 97, 121, 134], dispatching drivers [27, 47, 65, 95], or matching ride-sharing passengers [6, 73, 113]. See [3] for a survey.

Some papers provide detailed numerical simulations and heuristic algorithms. Vazifeh et al. [128] study the Minimum Fleet Problem, while Schilde et al. [114] provide heuristics for the dispatch and scheduling "dial a ride" problem.

Our contribution to this literature is two-fold. First, our findings in Chapter 6 provide a theoretical grounding to the popular re-optimization heuristic sometimes called *rolling-horizon*. Second, our simple dynamic matching framework can easily be constrained to incorporate only *realistic* match values in the context of ride-sharing. This could be used to study more in depth the value of waiting and optimization for different kinds of dynamic allocation problems.

Part I

Interface between Matching Technology, Market Heterogeneity and Match Frequency

Chapter 3

The Effect of Match Run Frequency on the Number of Transplants and Waiting Times in Kidney Exchange

3.1 Introduction

Kidney exchange, also called kidney paired donation (KPD), enables candidates with incompatible living donors to obtain transplants from other living donors, such as nondirected donors (NDDs) or donors belonging to other incompatible pairs [28, 39, 64, 85, 89, 90, 96, 99, 101, 109, 112, 116].[1] KPD programs perform match-runs that use optimization to find cyclic exchanges among incompatible pairs, and chains initiated by a NDD. Intuitively, an important factor that impacts the number of transplants is the size of the pool, which may be affected by the length of time between match-runs.

While the timing for deceased organ allocation is determined by the availability of organs, the timing of match-runs in KPD is more flexible. Longer intervals between match-runs allow for the accumulation of more pairs in the pool and may allow more potential matches. However, we will see that there is an important difference between pool size and composition. A larger pool of patient-donor pairs who have not previously failed to match provides many more matches than an equally large pool that includes many pairs for whom no matches were accomplished in previous match-runs, and this will reduce the benefits of delaying match-runs in mature pools that contain many hard-to-match pairs. Furthermore, infrequent match-runs may also slow down the complex process of identifying matches and carrying out transplants. This problem is amplified by the large fraction of proposed virtual matches that fail because of immunological, logistical, and other reasons [59, 64].Furthermore, additional time on the waiting list is undesirable for candidates [84].

National KPD programs in the United Kingdom, The Netherlands, Australia, and Canada conduct infrequent match-runs. In the United Kingdom, The Netherlands, and Australia, a match-run is conducted every 3 months and in Canada every 4 months [40, 56, 57, 69, 80]. 15-19 In the United States, KPD programs typically match very frequently: multicenter programs such as the Alliance for Paired Donation (APD) and National Kidney Registry match daily, United Network for Organ Sharing finds matches weekly, and also single-center programs such as Methodist at San Antonio (MSA) search for matches whenever a new

[1] See section 1.2.1 for an introduction on kidney exchange.

pair becomes available. In the United States, competition among KPD programs to produce transplants may have incentivized programs to perform match-runs at high frequency, which raises a major concern that such frequent matching may lead to fewer transplants [59]. In particular, matching frequently may lead to inefficient use of easy-to-match donors and missed opportunities for the most sensitized candidates. This article studies the impact of matching frequencies on the number of potential transplants and on the average waiting time to transplant in a pool of candidate-donor pairs. While matching frequency may affect outcomes by changing the pool size, other factors that determine pool size include acquisition rate and departure rate. This article further explores how these factors impact the fraction of the pool transplanted. We use the set of enrolled pairs from both the APD over a 9-year period and the MSA over a 3.5-year period. The MSA and APD provide us with 2 distinct, nonoverlapping datasets with very different pools of participating pairs (eg, number of: blood type O donors, easy-to-match pairs, NDDs, and compatible pairs) and different operational practices that significantly impact the connectivity of the respective pools. These different datasets allow us to evaluate the effects of match-run frequency in very different environments, thus providing a robustness check for the policies studied.

3.2 Methods

3.2.1 Data

The APD data consist of the characteristics of all incompatible pairs, NDDs and patients without a donor who enrolled in the APD registry between January 1, 2007 and August 11, 2016, including 1571 incompatible pairs and 50 NDDs. The MSA data consist of similar data for pairs entering from July 1, 2013 to February 1, 2017 including 592 pairs and 4 NDDs. The pairs' ABO distribution is given in Table 1 (for patients with multiple donors we select 1 donor randomly for this distribution) and Pool panel reactive antibodies (PRA) distribution in Table 2.

pABO \dABO	AB	B	A	O
AB	0.32	0.45	0.7	0.7
B	1.15	2.16	6.62	4.33
A	1.4	3.95	9.42	8.53
O	2.42	8.4	28.64	20.81

pABO \dABO	AB	B	A	O
AB	0	0	1.01	1.35
B	0.68	1.35	1.05	5.57
A	1.35	4.05	7.26	16.89
O	0.68	7.43	21.79	26.52

Table 3.1: Pairs type distribution in the Alliance for Paired Donation (APD) and Methodist at San Antonio (MSA). Left: APD, Right: MSA

	0-90	90-98	98-100
APD	0.5845	0.0946	0.3209
MSA	0.6416	0.1286	0.2298

Table 3.2: Pool Panel Reactive Antibodies (PRA) distribution in the MSA and APD pools.

The compatibility between a patient and a donor is determined by their blood types and a virtual crossmatch test, which compares the patient's antibodies (as entered by the patient's transplant center) and the donor's human leukocyte antigen. In addition to the virtual cross-match, transplant centers perform a crossmatch to verify compatibility. Finally, proposed offers fail to culminate in transplants for a variety of reasons that we model with failure rates described below. The 2 main types of failures occur due to positive crossmatch and rejection of the proposed donor by the recipient's center [59].

3.2.2 MSA vs APD data composition

This study does not allow us to compare the efficiency of the APD and MSA. Part of the value of these 2 datasets is the real differences in the connectivity of the pools' compatibility graphs between APD and MSA (ie, to the extent to which pairs are likely to be able to exchange with others in the pool). For example, the MSA dataset does not have to take into account discretionary exclusion criteria by different transplant centers as is done in the

APD, which lowers the connectivity of the APD pool. MSA further allows higher mean fluorescence intensity (MFI) cutoffs than APD centers allow, which translates to more compatibilities between donors and patients and hence a more connected compatibility graph (ie, more possibilities of donation from 1 pair to another). In addition, the MSA dataset has more compatible pairs participating compared to APD and no "selection" in that all pairs participate in the MSA system, whereas the APD loses easy-to-match and compatible pairs due to internal matching outside the APD. MSA also has a higher percentage of pairs with multiple donors than APD, again increasing possibilities of donation from 1 pair to another. Because of the substantial differences between the 2 pools, the computational experiments we conduct on each pool will be informative by providing "within-experiment" comparisons of different match frequencies, etc. within each pool. The 2 very different pools will in turn allow us to explore the impact of match-run frequency in the presence of differences such as arrival rate, composition of the pool (ie, number of blood type O donors, easy-to-match pairs, number of NDDs, number of compatible pairs) and departure rate, as well as the connectivity of the pools discussed above.

3.2.3 Optimization

For each match-run, we execute the matching algorithm on the available pool of incompatible pairs and NDDs to find a maximum "weighted" solution. One type of sensitivity analysis we conduct is to vary the priorities assigned to different patients based on their level of sensitization. We tested 3 strategies (S1, S2, S3) that use different weights assigned to a given patient's transplant based on the patient's calculated PRA (cPRA) (Table 3). Waiting time was not prioritized in these strategies because in the steady state, this will not change the average waiting time, due to Little's Law [131]. Our matching algorithm allows for cycles/loops of length at most 3 and chains of any length. The last donor of a chain becomes a "bridge donor" who continues the chain in the next match-run. A chain is terminated if the bridge donor remains in the pool for 3 months, by assuming the bridge donor donates to a patient on the deceased donor waiting list who does not have an incompatible donor. We do not include the chain-ending transplant in our analysis in order not to bias the outcomes of transplanted patients in the pool (eg, the large number of patients on the deceased donor waiting list allows ending chains with very highly sensitized patients).

cPRA	S1	S2	S3
98-100	1.05	2	10
90-98	1.05	1.5	5
80-90	1.05	1.05	2
0-80	1	1	1

Table 3.3: Base weights assigned by strategies 1 through 3 (S1-S3). Strategies S1, S2, S3 set the weight only according to the recipient's calculated panel reactive antibodies (cPRA).

3.2.4 Simulation Design

For each set of parameters, we run Monte Carlo simulations for 50 iterations. In each iteration we simulate the arrival of 5000 pairs and a small fraction of NDDs. The goal is to analyze the steady state that is reached by having a departure rate as described below. Each time period (number of days) we sample from the data pairs and NDDs with replacement according to some fixed arrival rate. In the base case, a pair or an NDD joins the pool every 2 days. We also simulate the departure of pairs and NDDs from the pool without being matched as observed in the data. For the APD, we set the number of NDDs that join the pool (during the arrival of 5000 pairs) to 160. The base case departure rate for the APD data is estimated using a Cox model, and on average a pair or NDD remains in the pool for 420 days (this rate varies only slightly across different types of pairs). In particular, each pair or NDD leaves the pool with probability 1/420 per day independently for reasons other than a match within the pool. We further conduct sensitivity analysis both on arrival rates and departure rates. For the MSA data we do not have good estimates and set the base case average stay in the pool to 800 days. This was chosen to be larger than the estimated departure rate of the APD because pairs at the MSA do not enroll in competing exchange programs (but here too we conduct a sensitivity analysis). Due to the very small number of NDDs in the MSA data, we restrict attention only to pairs and thus assume there are no chains when using MSA data (however, we also conducted simulations with chains and found similar qualitative results). While the simulation is run until all pairs have arrived, statistics are measured only for pairs that arrived after the 100th pair arrived to decrease the biases at the "beginning" of the simulation and capture steady-state results. Simulations were run for different matching frequencies. We model the failure rate of match offers being converted to transplants. In APD, after the matching algorithm identifies a match, each candidate's center has up to 1 day to accept the offer or not. Some offers are turned down by centers for nonimmunological reasons despite the fact that they should have preselected only donors that are acceptable for their patients. If all offers within a chain or a loop are accepted, centers are asked to conduct the actual cross-matches and exchange more extensive patient and donor medical records. Using estimates from Fumo et al. [59], 1-way offers are estimated to be rejected 27% of the time; the involved pairs are returned to the pool after keeping them inactive for the time it takes to add 1 pair to the pool. Actual crossmatch failures were set to occur with a probability of 38% for patients with cPRA >90 and 10% for all other patients. The transplants in the chain were conducted until the first failure. Two types of models are simulated for realizing and resolving failures motivated by practices in single and multihospital exchange programs: No-delay model: Failures are realized immediately, allowing for instantaneous reoptimization. This model concentrates on the impact of waiting between match-runs. This model is equivalent to reoptimizing immediately after every failure over the entire pool. (A similar reoptimization approach is adopted in single-center programs such as MSA and partially in national programs such as the United Kingdom and The Netherlands.) Delay model: Failures are resolved over time as observed in US multihospital KPD registries. In our simulations, a pair becomes inactive during an offer stage until 2 days elapse and during a crossmatch stage it is inactive until 7 more days elapse. Patients were considered transplanted if there were no failures in the chain or cycle. All the simulations using MSA data assume the no-delay model, as MSA is a single-center program that can reoptimize immediately after some failure occurred. In fact, most simulations are conducted under the no-delay model in order to evaluate the impact of policies on transplants in "best-case" scenarios.

3.3 Results

The measured simulation outcomes are the fraction of patient-donor pairs transplanted and the average waiting time to transplants experienced by the candidates over the entire study. Simulations were run with the intervals between match-runs of 2, 4, 7, 14, 30, 60, and 120 days.

3.3.1 Impact of match-run frequency under different prioritization strategies

We first test the effect of varying the interval between match-runs under different prioritization strategies. Figure 1 describes outcomes under the delay model (failures are resolved over time) and Figure 2 describes outcomes under the no-delay model (immediate resolution of failures).

The top left plots of Figures 1 and 2 show that under both models, regardless of the prioritization strategy, the average fraction of transplanted patients decreases as the interval between match-runs exceeds 7-14 days. Top right plots of these figures show that the average waiting time increases as the interval between match-runs increases. The effect of change is more significant in Figure 1 than in Figure 2 because the time required to resolve failures harms the match rate and the waiting time. This is most evident in the bottom right plots, which show the average waiting time for overdemanded (easy-to-match pairs); these pairs can match immediately when failures are resolved immediately (explaining why the average waiting time is about half the length of interval between match-runs), but have to wait significantly more to be part of a cycle or a chain when failures take time to resolve. Some prioritization may help the most highly sensitized patients (middle plots in Figures 1 and 2). However, in those cases fewer low sensitized patients and fewer underdemanded pairs

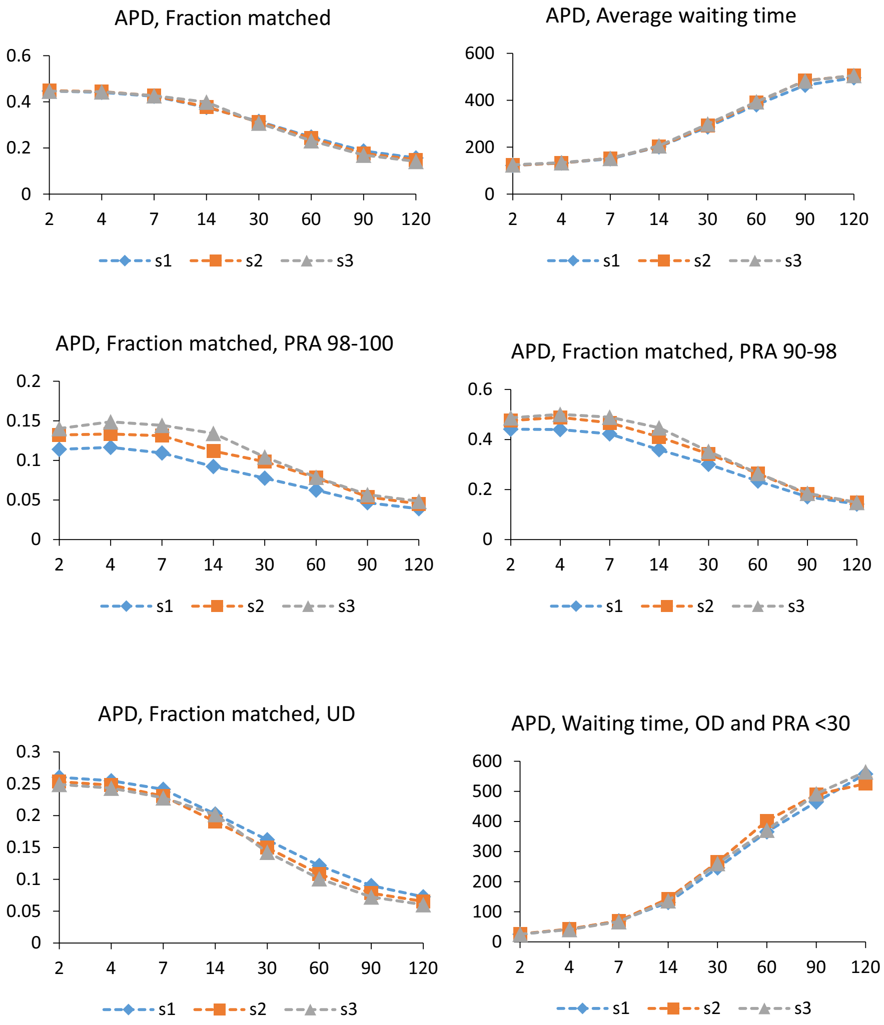

Figure 3-1: Statistics under the delay model using APD data. The x-axis represents the time interval between 2 match-runs. Strategies S1-S3 are defined in the "strategies" section and Table 3. (Top left) Fraction of matched pairs. (Top right) Average waiting time. (Middle) Fraction of match patients with high PRA. (Bottom) Fraction of underdemanded pairs matched and the average waiting time of overdemanded pairs with low PRA. APD, Alliance for Paired Donation; OD, overdemanded; PRA, panel reactive antibodies; UD, underdemanded

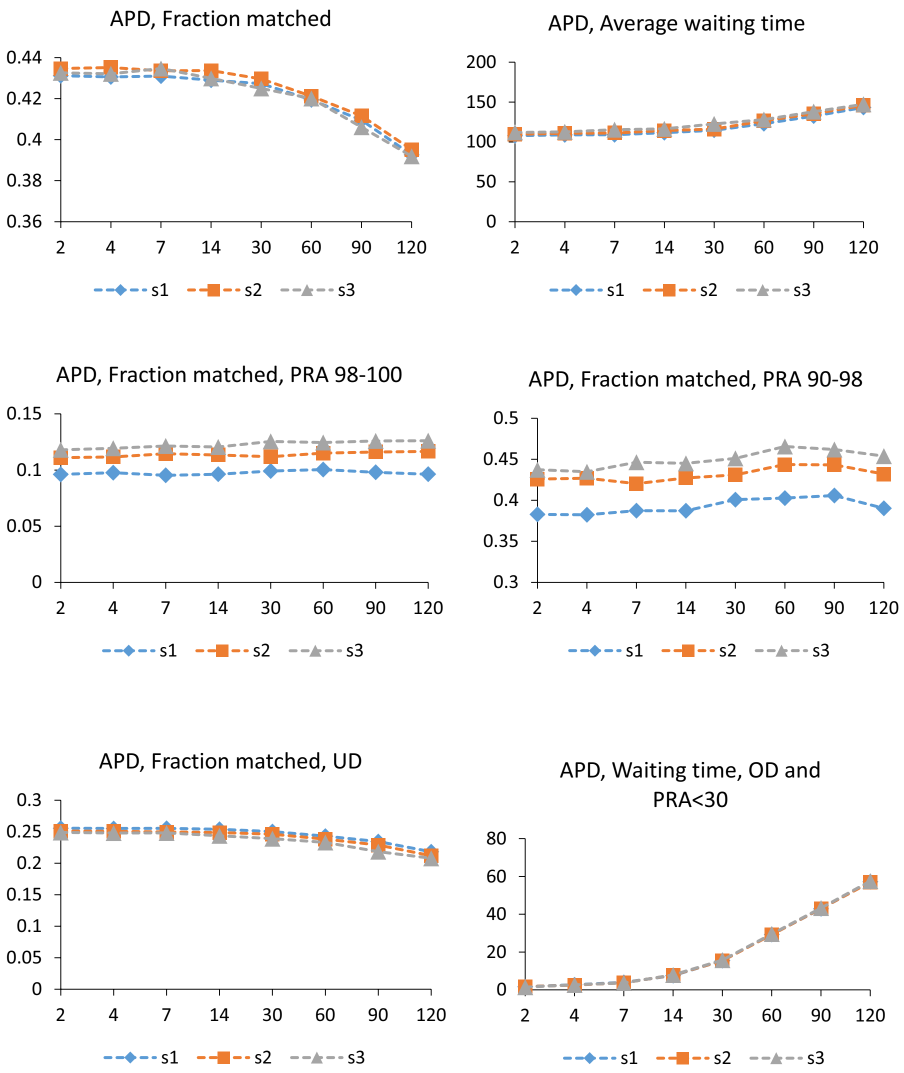

Figure 3-2: Statistics under the no-delay model using APD data. The x-axis represents the time interval between 2 match-runs. Strategies S1-S3 are defined in the "strategies" section and Table 3. (Top left) Fraction of matched pairs. (Top right) Average waiting time. (Middle) Fraction of match patients with high PRA. (Bottom) Fraction of underdemanded pairs matched and the average waiting time of overdemanded pairs with low PRA. APD, Alliance for Paired Donation; OD, overdemanded; PRA, panel reactive antibodies; UD, underdemanded

are transplanted (patients-donor pairs with ABO types X-Y are under-demanded if they need a scarcer kidney than they are offering in exchange (ie, if X is ABO compatible with Y but Y is not ABO compatible with X); that includes O-A, O-B, O-AB, A-AB, B-AB). We report very similar qualitative results using MSA data (Figure 3). While the qualitative observations are similar, there are large differences between the MSA and APD (for instance, 11% difference in the fraction of matched pairs, Figures 2 and 3, under the no-delay model). This difference in the fraction of matched pairs is due to differences in the connectivity of the pools. Note that in Table 1, the ratios of O donors to O patients are .57 and .89 in the APD and MSA pools, respectively. Also, 34.4% of APD pairs contain an O donor, whereas 50.3% of the MSA pairs have an O donor, which strongly impacts the difference in fraction matched and the difference in fraction of underdemanded matched. Thus, independent of the pool connectivity, matching infrequently does not increase the fraction of matched pairs.

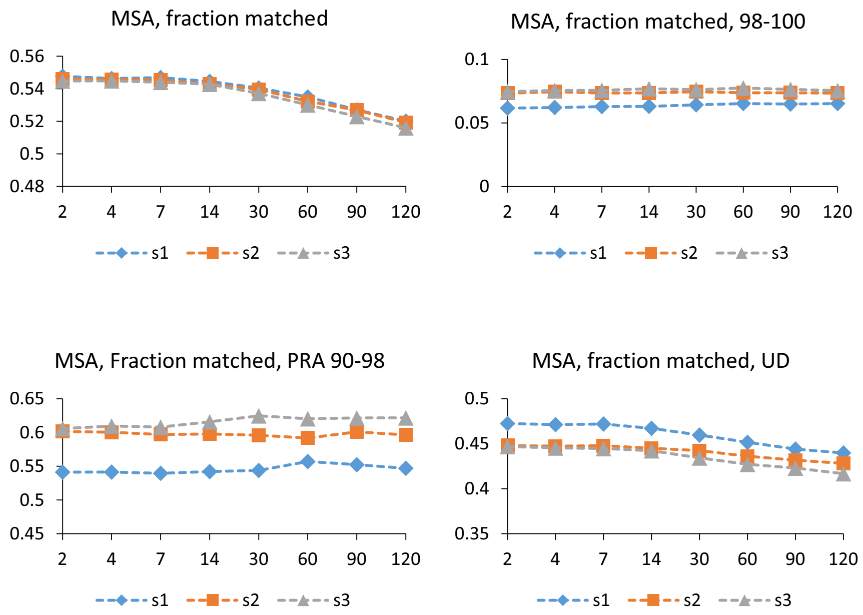

Figure 3-3: Statistics under the no-delay model using MSA data. The x-axis represents the time interval between 2 match-runs. Strategies S1-S3 are defined in the "strategies" section and Table 3. (Top left) Fraction of matched pairs. (Top right) Fraction of matched patients with PRA 98-100. (Bottom left) Fraction of match patients with PRA 90-98. (Bottom right) Fraction of matched underdemanded pairs. MSA, Methodist at San Antonio; PRA, panel reactive antibodies; UD, underdemanded

We emphasize that a low matching frequency under the delay model is not very practical since this requires the KPD program to wait with failed matches (without reoptimizing) until the next match-run. We next focus on the no-delay model where failures are resolved immediately (if infrequent matching is not helpful for this model, it is not expected to be helpful under the delay model). For simplicity we also present the next results only for strategy S2 as we find no qualitative differences between the 3 strategies in the simulations. For clarity we present the next results only under strategy S2 as we find no qualitative differences in the simulations.

3.3.2 Varying arrival and/or departure rates

We conduct sensitivity analysis on the arrival rate of new pairs (base case is 1 arrival every 2 days = 1 period) and report in Figure 4 simulation results under the no-delay model when a pair arrives every t periods (t = 1, 2, 4, 7, 14). For each arrival rate, the fraction of transplanted pairs does not increase as the interval between match-runs increases. However, the greater the arrival rate, the greater the fraction of transplanted pairs and the lower the waiting time (note: the larger the t, the lower the arrival rate). The bottom plots provide a different view of the results. Note that the arrival rate is a major factor determining the fraction of transplanted patients, whereas the matching frequency plots essentially coincide (except the lowest frequencies). The lower the arrival rate the more benefit there is to increase arriving pairs. This benefit is minor for very high arrival rates (after 350-700 pairs annually).

We also varied the average time a pair remains in the pool in MSA and APD (x = 420, 800, 1000). The results (Figure 5) show that as the departure rate becomes large (1/x), the smaller the fraction of pairs that are transplanted. However, for every departure rate, matching frequently does not harm the fraction of matched pairs. Like the impact of arrival rate, decreasing departure rate increases the fraction of transplanted patients.

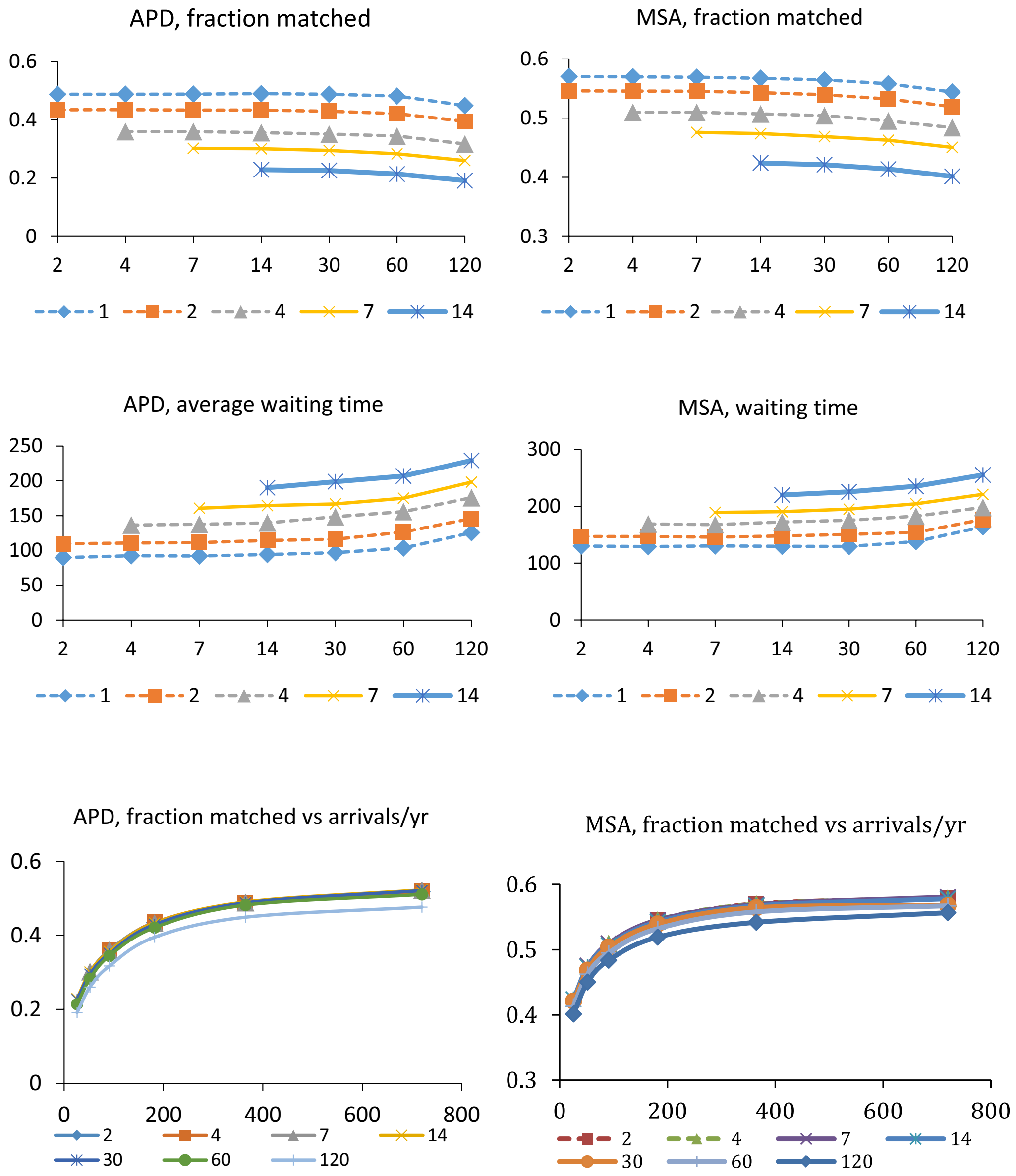

Figure 3-4: Sensitivity analysis over arrival rates under the no-delay model and strategy S2 in APD and MSA datasets. In the first 4 figures, each curve represents an arrival rate by the number of days between the arrival of each pair or NDD, and the x-axis represents the time interval between 2 match-runs. The 2 bottom figures are similar to the top 2 figures only the x-axis represent the number of pairs arriving per year and each curve represents the interval length between 2 match- runs. APD, Alliance for Paired Donation; MSA, Methodist at San Antonio; NDD, nondirected donor

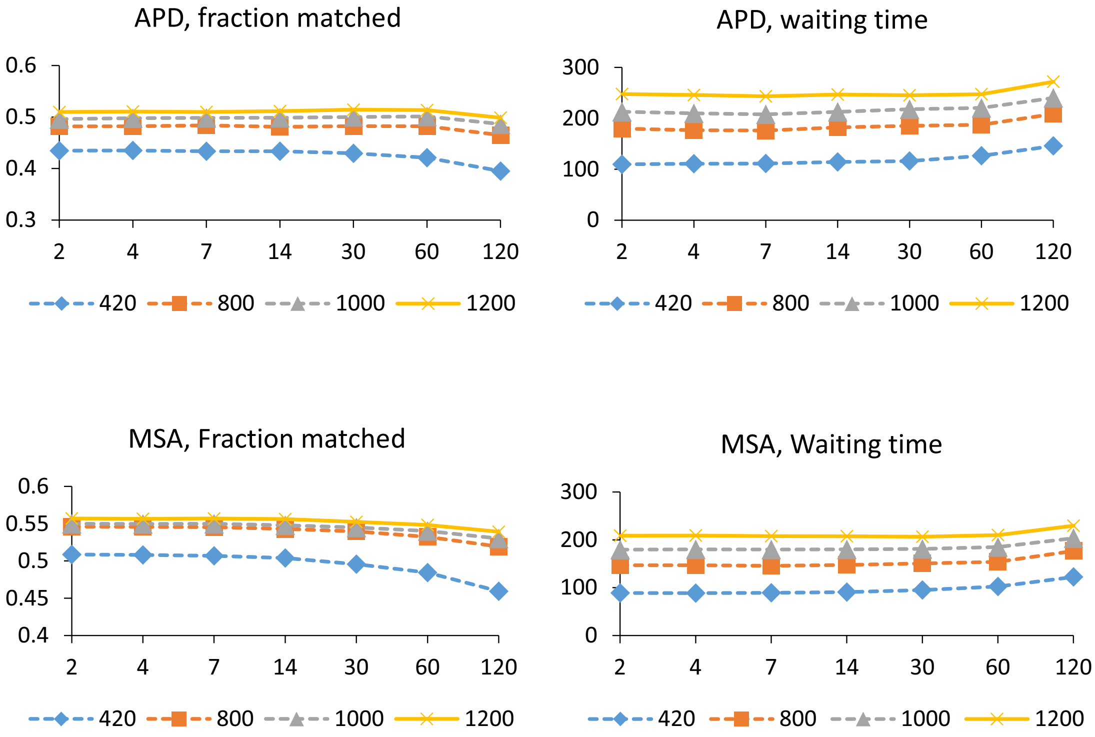

Figure 3-5: Sensitivity analysis over departure rates under the no-delay model and strategy S2 in the APD and MSA data. The x-axis represents the time interval between 2 match-runs. Each line represents a different departure rate where the numbers correspond to the average number of days a pair remains in the pool without being matched. (Left) Fraction of matched pairs. (Right) Average waiting time. APD, Alliance for Paired Donation; MSA, Methodist at San Antonio

3.3.3 Varying practical constraints: NDDs and cycle length

We ran similar simulations to explore different constraints. Simulations assuming no NDDs in the APD result in similar patterns (Figure 6 left). We relaxed the maximum cycle length to allow for 4-way cycles, and, while the fraction of patients transplanted increases, frequent matching does not harm the fraction of transplanted pairs (Figure 6 right).

3.3.4 Match efficiency

One indicator of the matching efficiency of a KPD program is the fraction of blood type O donor kidneys that are transplanted into blood type O patients (intuitively, in a very large

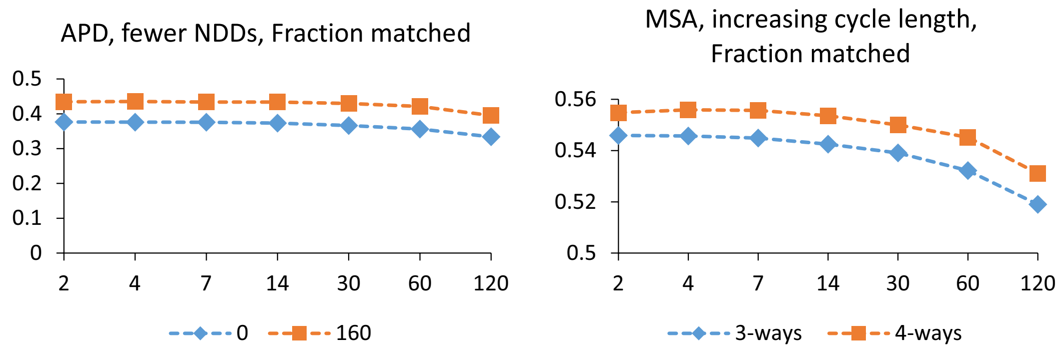

Figure 3-6: Different number of NDDs (left) and different cycle length (right). Both simulations use the no-delay model and strategy S2.The x-axis represents the time interval between 2 match-runs. Testing base case arrival of NDDs (160) and no NDDs at all. (Left) Testing both 3-way and 4-way cycles. (Right) Average waiting time. APD, Alliance for Paired Donation; MSA, Methodist at San Antonio; NDDs, nondirected donors

pool all blood type O donor kidneys would be transplanted into blood type O patients). Figure 7 shows that using longer match-run intervals does not increase this measure under different prioritization strategies and different arrival rates. However, increasing the arrival rate results in a higher blood type O match efficiency.

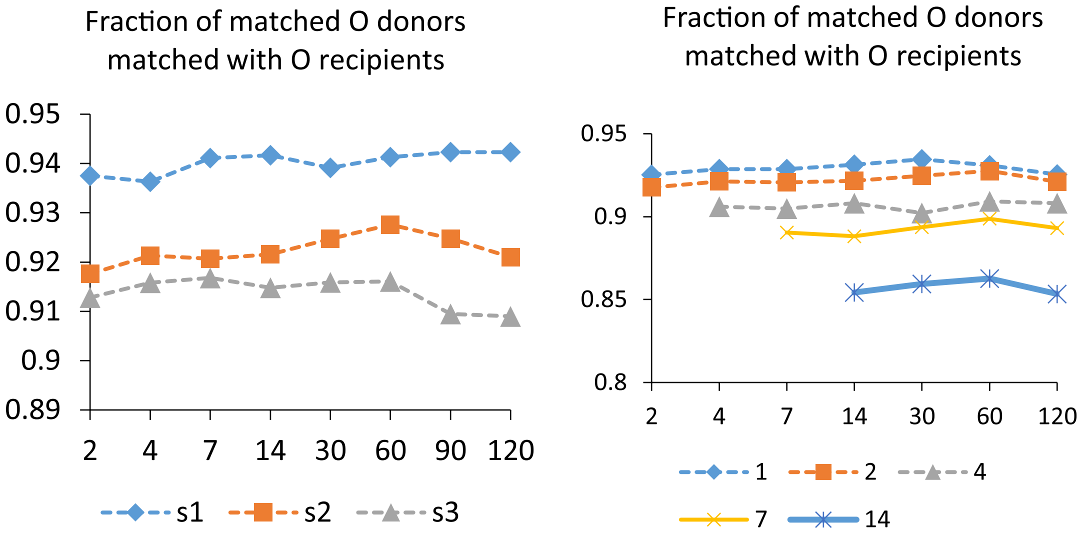

Figure 3-7: Fraction of matched O donors that are matched with O patients. The left plots different prioritization strategies and the right plots different arrival rates for strategy S2. The x-axis represents the time interval between 2 match-runs.

3.4 Discussion

As KPD has become more widely used, the databases of patients and donors have grown rapidly and they contain a large fraction of highly sensitized patients [78]. It is therefore important to evaluate the effect of increasing the pool size in order to create more opportunities for these patients. To do so we vary the match-run frequencies (which can be determined by the KPD program) but also vary the exogenous arrival rate (which is a consequence of participation and collaboration). Using the accumulated patient/donor pool at the APD and MSA databases, we modeled running a matching algorithm, making match offers, accounting for rejected offers, and simulated laboratory cross-matches. We varied the match-run frequency, the arrival rate to the pool, and the departure rate from the pool. Sensitivity analyses were performed for pool connectivity, priorities assigned to highly sensitized patients, failure rates, use of NDDs, and chain/cycle length. We find that matching frequently does not reduce the fraction of the pool matched. Importantly, however, the fraction of the pool transplanted does increase as either the arrival rate increases or the departure rate decreases. In fact, increasing arrival rate is the most important modifiable factor to increase the fraction of the pool transplanted, particularly for low arrival rates. While the exact numbers might differ across datasets, the general principles will still hold. These results help illustrate why the size of the pool is not by itself a good indicator of the fraction of patients who can be transplanted. A large arrival rate means a large pool with many matchable pairs, while a low fraction of transplantable patients can also produce a large pool, but of hard-to-match pairs. The lesson for the United States is that KPD programs should consider efforts to collaborate to increase their arrival rate. While waiting 1-2 weeks between match-runs does not reduce the fraction of pool transplanted, using this time to clarify competing matches for easy-to-match pairs by using different strategies may help to achieve predetermined goals such as more transplants for hard-to-match patients. For non-US KPD programs that perform match-runs less frequently than every month and have nonnegligible departure rates, it may worth experimenting with more frequent match-runs. Finally, non-US programs with a low acquisition rate and low match rate may benefit from international collaboration to increase their acquisition rate. This is likely to have a much larger impact on the fraction of the pool they match than does match frequency. The logistics of international exchange currently serve as a barrier to broader collaboration, and strategies to overcome these barriers should become an active area of research. Intuitively, matching frequently does not harm the fraction of transplanted patients because both underdemanded pairs (such as O-A patient-donor pairs) and highly sensitized patients accumulate in the pool. When, for example, a patient-donor A-O pair arrives, if the patient is low sensitized the pair can match immediately with an O-A pair, which is an efficient match. If the A patient is highly sensitized and cannot match with any of the accumulated O-A pairs, it is also unlikely that this patient can match any pair arriving in the near future, so it does not increase the total number of transplants to postpone matching other pairs until a donor compatible with this patient arrives. Put differently, when the departure rate is low, many hard-to-match pairs accumulate in the pool and so waiting with a newly arriving easy-to-match pair is unnecessary since it is likely to match to one of the already present hard-to-match pairs. And when the departure rate is high, matching infrequently will result in many departures of easy-to-match pairs. Moreover, that a low arrival rate yields a match rate below the maximum is a result of both departures of unmatched pairs and suboptimal matching, which would not have happened in a thicker pool. For example, with a low arrival rate, some O donors might match A patients, but with a large arrival rate such A patients could be matched by A donors and such O donors could match O patients. Finally, it is worth noting Little's Law [131], which states that in steady state, the average pool size equals the arrival rate multiplied by the average time a pair remains in the pool (whether matched eventually or not). So for a given arrival rate, one may artificially increase the pool size by waiting between match-runs. However, this can increase the average waiting time as well, which may lead to more departures. Since matching at some KPD programs in the United States is more aligned with the delay model, it is reasonable to use high matching frequencies. Our results suggest that even under the no-delay model (where failures are resolved before the next match run), high matching frequency is a reasonable strategy. Our approach is very different from that of Segev et al. [117] In our approach the pool evolves dynamically with arrival and departures modeled over time. Segev et al. [117] examined a large static pool, and compared matching pairs sequentially using a "first-accept" approach, to optimizing over the entire pool (a single match-run). Matching frequently in our (steady-state) model is thus different from their first-accept approach. The key difference lies in the composition of the pool: When we match frequently and seek to match an easy-to-match pair, our evolving pool contains mostly hard-to-match pairs, while in their model, which considers a single matching cycle, there could be many other easy-to-match pairs in the pool and matching easy-to-match pairs with each other is often inefficient. While we model departures, our findings hold even for very low departure rates. Finally, we always optimize while prioritizing hard-to-match pairs, regardless of the matching frequency.

KPD programs vary in the priorities they use. Our findings suggest that while prioritization of highly sensitized patients increases the percentage of these patients transplanted, it does not significantly increase the total number of transplants. However, guidelines for how to prioritize pairs can come from studying unmatched departures.

Some strategies used by MSA affect matching frequency and are driven by other factors. Donors may have strong preferences over when to donate, so it is important to prioritize donors whose window for donation closes soon. Moreover, compatible pairs should be given high priority, otherwise they may choose to depart to conduct a direct transplant. These strategies are consistent with matching frequently.

This study has limitations. Only a limited number of strategies are considered and some other strategies may perform better. Strategies that consider the future may have benefits over strategies that optimize in the current pool [45, 48]. However, this is unlikely when the arrival rate is high, since it will be possible to match easy-to-match pairs upon arrival to hard-to-match pairs due to the accumulation of the latter. Also, while patient data are taken from actual KPD registries, we made simplifying assumptions that may weaken our conclusions. We assumed failure rates are independent, and assumed a steady influx of pairs into the database. However, we emphasize that while we report only a representative set of simulations, we found similar qualitative findings under a much broader set of strategies and with lower failure rates. Also, while frequent and infrequent match-runs result in a similar fraction of matched pairs, matching infrequently may allow an increase in match quality. Additionally, departure rates in our simulations are identical for all pairs. If frequent matching for a given KPD program is a good strategy with identical departure rates, it would remain a good strategy also when easier-to-match pairs depart faster than harder-to-match pairs since these pairs match quickly in our simulations. Finally, some departures are due to transplants, which are good outcomes. Thus, the reasons for departures from a KPD pool should be studied. We also do not explicitly study competition between KPD programs. However, we predict the following effects: When patients do not cross-register, the existence of multiple programs reduces the arrival rate of each; when some patients cross-register, departure rates may be influenced by match rates at competing programs. So matching frequency, by affecting the match rate, affects departures and arrivals at competing programs, and the overall chance of a pair to match should be further studied. In summary, while we do not find that frequent match-runs result in fewer transplants, we do find that increasing arrival rates and decreasing departure rates improves both the fraction of matched pairs and waiting times. So while the fraction of matched patients (and their waiting times) may be harmed by competition among KPD programs, it is unlikely due to the high frequency of match-runs, but rather due to low arrival rates of pairs and high departure rates.

Chapter 4

Cycle Length, Priority Schemes and Matching Frequency

4.1 Introduction

Kidney exchange pools currently experience a low arrival rate, and they contain a large fraction of highly sensitized patients (Ashlagi et al. [14]), who may need to wait a very long time to find a tissue-type compatible donor.[1] As a result, many patient-donor pairs depart the pool without being matched [2]. The length of stay of pairs varies substantially from a few days to several months depending on patient and donor characteristics.

Currently exchange programs periodically search for allocations, i.e. sets of disjoint exchanges, that maximize the number of pairs matched. at different frequencies. One of the major decision they are facing is how frequently to look for these allocations.[3] On one hand, waiting increases the chance that some pairs depart unmatched (especially pairs with low sensitized patients); on the other hand, clearing the market too frequently may reduce the number of transplants (especially for highly sensitized patients) by making suboptimal matching decisions.

In this paper, we take a data-driven approach to model the dynamics of kidney exchange, and study how the clearing frequency along with the length of exchange cycles impact the number of matched patients over time. We analyze a rich data set from a large exchange program (see Section 4.2 for details) to understand the keys features of this dynamic market.

In order for a patient to be transplanted with a donor's kidney, they need to be blood-type compatible as well as tissue-type compatible. Tissue type (or HLA)-compatibility is determined by comparing the patient's antibodies to the donor's human leukocyte antigens. Thus patients with a high number of antibodies will be less likely to find a compatible donor. We say that such patients are highly sensitized, and this is measured with the Panel Reactive Antibody (PRA). A patient's PRA captures the likelihood the patient is HLA-incompatible with a donor chosen at random in the population.

Ashlagi et al. [14] find that the percentage of highly sensitized patients (PRA above 95)

[1] This is partially since hospitals match internally their easy-to-match pairs and only register their hard-to-match ones Ashlagi and Roth [13].

[2] There could be a few reasons for such departures: the pair finds a match in another pool or from a deceased donor; the patient becomes too sick to transplant; the donor becomes unavailable.

[3] See section 3.1 for a discussion of the different match frequencies in KPD programs worldwide

in the pool is significantly higher than what previous studies have assumed to support earlier theoretical findings [107, 111]. We confirm these results with data from three different KPD programs in the US. Figure 4-1(left) plots the PRA distributions of patients enrolled at the NKR, APD and MSA. Note that blood-type compatibility is not incorporated in this aggregate PRA distribution. Figure 4-1(right) provides the same distributions for patients belonging to blood-type compatible pairs, who can match with each other if they are tissue-type compatible. These distributions can be roughly viewed as bimodal; note that among blood-type compatible pairs there are more highly sensitized patients than low sensitized ones.

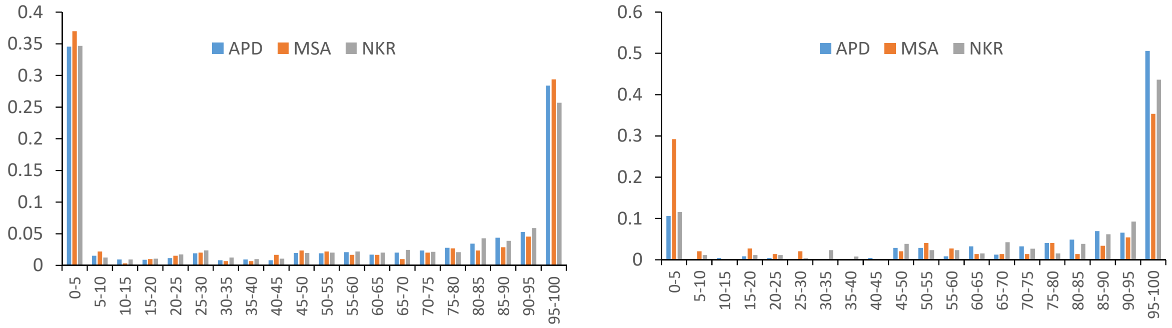

Figure 4-1: PRA distributions of patients enrolled at NKR (1/2012-12/2014), APD (1/2007-8/2016), and MSA (7/2013-2/2017). Left: all patients. Right: patients belonging to blood-type compatible pairs.

This bimodal distribution leads us to split patient-donor pairs into two categories: we will say that a patient-donor pair is easy-to-match if the patient has a PRA lower than 95. Conversely, we will say that a pair is hard-to-match if the patient has a PRA above 95. We find that even though hard-to-match pairs stay in the pool much longer, they still have significantly less matching opportunities than easy-to-match pairs.

This raises the following questions: given the dynamics of kidney exchange pools, can we understand how different priority and waiting mechanism impact different sub-populations of pairs? In particular can we improve the likelihood that a highly sensitized patient gets matched, and at what cost to the other patients? In this paper, we use our empirical findings to develop a dynamic model of kidney exchange. Using theoretical analysis as well as data-driven simulation we then study the performance of a class of simple policies that, similar to the current practice, periodically finds allocations. We compare different matching policies that determine when to match and how to prioritize.

In particular, we analyze an algorithm, hereafter called *b-batching*, which "waits" to receive b new pairs before it searches for a maximum size allocation in the existing pool. Motivated by current practice, we will focus on small values of b[4].

We consider an infinite horizon discrete time model where one pair arrives at every timestep. Each arriving pair is sampled from a bi-modal distribution independently. We model the length of stay of pairs as a Geometric random variable whose mean depends on its type. As we show in section 4.2, on average, highly sensitized patients stay in the pool longer. We model this arrival/depature dynamic along with the matching process as a Markov chain. Because our focus is on helping highly sensitized patients, we will compare policies based on the fraction of these patients that are matched in the long run (steady-state). We compare the performance of b-batching policy to the special case of $b = 1$, which we call the *greedy policy.*

We first study the setting in which allocations include only cycles of length 2. We show that, compared to the greedy policy, the benefit of b-batching is not significant (Theorem 1 and Remark 1). In contrast, we show that when cycles of length 3 are allowed, batching can result in considerable gain over greedy (Theorem 2 and Remark 2). This result highlights the fundamental difference between the structure of 2-way and 3-way cycles, and the importance of modeling heterogeneity which allows us to give different priorities to different types. From a design perspective, it implies that some waiting with the easy-to-match pairs can help hard-to-match ones. Particulary, when an easy-to-match pair arrives that is compatible with a hard-to-match patient in the pool, some waiting increases the likelihood that these two pairs can be part of a 3-way cycle.

In our numerical study, we simulate the process by first using a survival model to estimate departure rates, and then sampling patient-donor pairs from a clinical dataset. Our

[4] Intuitively, for large b, the loss due to departure will outweigh any potential benefit of waiting. As shown in our simulation result, practical value of b is between 1 and 10.

simulation results confirm the insights developed in our theoretical analysis that certain form of batching results in a significant gain in matching hard-to-match patients. However, when restricted to only 2-way matching, batching has no practical impact. We also discuss the impact of batching on the match rate for easy-to-match patients. Further, we extend our simulation study to include all blood-types and discuss how that impacts the performance.

Matching highly sensitized patients is one of the main challenges in kidney exchange. Giving priorities to these patients in the case of ties has been proven to be helpful (as we also show in Section 4.5.3 - Fig 4-10). The dynamic nature of matchings creates another dimension to prioritize hard-to-match pairs by making others “wait”. In this paper, through data-driven modeling, theoretical analysis, and simulation using clinical data, we show that (i) Making low sensitized pairs wait is only helpful if we allow for 3-way cycles in addition to 2-way cycles. (ii) Such a policy, even though beneficial for hard-to-match pairs, harms the matching of low sensitized patients, as more of them will leave unmatched.

Our results may be of independent interest to the literature on dynamic matching in random graphs. Kidney exchange serves well as an example for which we have distributional information on the underlying graphs, thus we can exploit this information to make analysis and prediction far more accurate than the worst-case analysis can do. We believe our average-case analysis can have implications beyond the kidney exchange and can be applied to other dynamic allocation problems with such distributional information.

4.1.1 Related work

Overall, as the literature on dynamic matching is developing, a clearer picture is emerging on the impact of waiting. In completely homogenous settings with no knowledge of departure times, waiting is ineffective Akbarpour et al. [4], Anderson et al. [9]. However, as the current paper witnesses, in models where heterogeneity in compatibility and/or preferences are incorporated, and more general forms of matching is allowed (e.g., 3-way cycles in addition to 2-ways) some positive impacts of waiting become apparent (See also [22]). Ünver [126] initiates the study of dynamic kidney exchange. He shows a closely related result to the static case under the assumption that no tissue-type incompatibilities exist. In particular, waiting is not an issue when only blood type compatibilities are considered. Our work deviates from his model by abstracting away from blood types and focusing on the tissue-type compatibility and the spareness of the compatibility graph. Further, our approach to study dynamic kidney exchange is combinatorial and is based on the structure of the underlying random graph while Ünver [126] takes a dynamic programming approach.

Roth et al. [103] first proposed a way to organize kidney exchange integrating cycles and chains. Logistical constraints required that cycles involve no more than 2 patient-donor pairs. Subsequent work suggested that a modest expand of infrastructure, that is allowing only slightly larger, 3- and 4-way exchanges would be efficient (Roth et al. [107], Ashlagi and Roth [12]) in large static pools. These studies assume either implicitly or explicitly that no tissue-type incompatibilities exist.

Awasthi and Sandholm [21] use online stochastic optimization methods to computationally study the allocation in kidney exchange using short cycles in a dynamic setting. Dickerson et al. [43] conduct computational simulations in the dynamic settings to understand the benefit of chains. Dickerson et al. [44] study dynamic optimization and propose an algorithm that assigns weights to different matches using future stochastic sampling. These studies use however dense compatibility graphs (they generate graphs according to Saidman et al. [111] which have been shown in Ashlagi et al. [14] to be very dense as opposed to clinical data). Further, [49] use a novel random walk approach to study the effectiveness of chains in a static random graph with hard and easy-to-match agents.

In a related line of work, Zenios [133] studied a dynamic kidney allocation model where both direct (2-way) and indirect kidney exchange are allowed. [5] The objective is to study the tradeoff between the (lower) quality of a deceased donor (that is assumed to be available immediately) and the cost of waiting for a direct exchange. The exchange model studied in this work ignores the tissue-type compatibility.

Another line of work in kidney exchange studies matching in the presence of failure. Many

[5] In an indirect exchange, the donor gives a kidney to the highest priority patient in deceased donor waiting list, and her intended patient gets the highest priority in the waiting list.

of the match offers given by the exchange programs fail to proceed to actual transplant for various logistical and medical reasons. A few recent papers study kidney exchange with failures, [31, 48, 88]. They use a probabilistic model for failure and assume that each compatible match is only viable with a given probability. They design algorithms to maximize the expected number of successful matches in static settings. Further, [48] computationally study this problem in a dynamic setting using dense graph models.

The problem of online matching (equivalent to our online scenario with only two-ways) arises naturally in information technology applications such as online advertising in which advertisements need to be assigned instantly to queries searched or webpages viewed by users. The study of online matching was initiated by Karp et al. [72], in which they analyze the problem in adversarial settings with no probabilistic information about the graph. Several follow up papers, studied the problem in settings that limit the power of the adversary. Goel and Mehta [61] studied the model in which the underlying graph has unknown distribution. Feldman et al. [54] noticed that in applications such as online advertising there is information about the graph structure, and they analyzed a model where the graph distribution belongs to a certain class. Jaillet and Lu [67], Manshadi et al. [81] studied the same problem with a general known distribution. Note that here we focus on one special class of distributions; however, unlike the computer science literature, we consider various regimes of waiting (and not just the online scenario).

Mendelson [86] analyzed the behavior of a clearinghouse in a dynamic market with prices in which sellers and buyers arrive over time according to a given stochastic process. Similar to our work, he considers a mechanism in which the clearing prices are computed periodically, and he studies the market behavior for different time (period) scales.

Since the first version of this work, a few other papers followed up this line of work. [4] and [9] studied dynamic barter exchange in a homogenous model where each 2 pairs are compatible with each other independently with a very small probability. Similar to our work, both of these papers are concerned with the impact of waiting before matching on the performance of the market. [4] only study 2-way matching. It explicitly models departures and focuses on loss rate as the measure of performance. They show that if the clearinghouse has information about when a pair departs then matching without waiting is suboptimal. However if pairs depart probabilistically (similar to our model) then greedy is almost optimal. [9] study 2-way, 3-way matching as well as chains. In their model, pairs never leave before getting matched. Therefore they use the average waiting as a measure of performance of a policy. [6] They show that in all of their settings the greedy policy (that matches without any waiting) is almost optimal.

Main differences between our work and theirs is that (i) we consider a heterogenous model which seems to be closer to the clinical data for kidney exchange (ii) we study both 2-way and 3-way matching, and we show that there is a fundamental difference between the impact of waiting in 2-way and 3-way matching. This difference crucially stems from having heterogeneity in level of compatibilities. We note our finding for 2-way matching is consistent with both papers. However, our finding for 3-way is different from that of [9], essentially because of having two types instead of one. (iii) We carry out an extensive simulation study using clinical data to validate our theory, and further study dynamic policies in more general settings.

4.2 Empirical findings

In this section, analyze a historical data from a major kidney exchange programs in the U.S.: National Kidney Registry (NKR). These findings will form the basis of modeling assumptions we make later in our theoretical and simulation study. From 01/01/2012 to 07/01/2014, 1162 incompatible patient-donor pairs joined NKR. This corresponds to an average arrival rate of 1.28 pairs per day. The objective of this section is to analyze the heterogeneity of agents in a Kidney Exchange system, and construct a data-driven model for Kidney exchange and derive properties of efficient matching mechanisms.

[6] However, they argue that in a model with departure reducing the wait time is aligned with reducing the loss rate.

Arrivals

Ünver [126] shows that because of blood-type compatibility issues, some pairs are on the "long" side of the market and it is therefore not possible to match all of them. They correspond to pairs where the donor is blood-type incompatible with the patient, and they are called *under-demanded* pairs. In this paper, we will not consider blood-type related compatibility structure. Therefore, most of our results in this chapter apply to the subset of the market that contains the self-demanded and over-demanded pairs where the donor is blood-type compatible with its intended recipient. Table 4.1 gives a summary of the blood-types of patient-donor pairs.

Patient \Donor	O	A	B	AB
O	19.2%	**27.0%**	**10.8%**	**1.9%**
A	7.7%	9.3%	4.8%	**1.7%**
B	4.0%	7.1	2.5%	**1.2%**
AB	0.6%	1.2%	0.3%	0.6%

Table 4.1: Blood-type statistics of pairs in NKR historic data. The under-demanded pairs are shown in bold.

Figure 4-2 shows how the difficulty to find a match relates to the pair PRA.

We remark that this PRA distribution is different from what we would see in the general population, because the pairs joining the pool are not uniformly random samples of the population. As we can see from Table 4.1, 529 blood-type compatible pairs join the pool. Most of these pairs are very likely to have highly sensitized patients [7] . Another reason that accounts for this biased sampling is the incentives of the hospitals to withhold their easy to match patients, so that they can match them internally, and to only enter their hard to match patients into the exchange program (see Ashlagi and Roth [13] for more details). Because this paper focuses on sensitization, in all the data analysis that follows, we consider only the 667 pairs that are not under-demanded.

Departing from the classical one-type model in the Kidney exchange litterature is a

[7] Think of all the pairs with O-patient and O-donor, those that join the pool are tissue-type incompatible, therefore more likely to be highly sensitized.

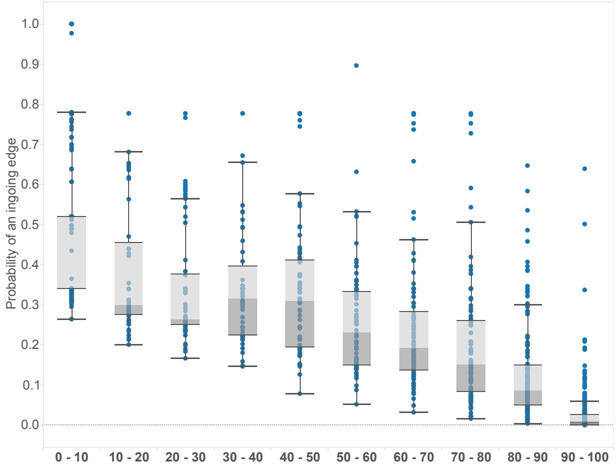

Figure 4-2: Probability of receiving an ingoing edge from a random pair in the system (i.e. ratio of total incoming edges to total number of pairs) as a function of the PRA. Each dot corresponds to a pair, the probability is computed as the ratio of the number of ingoing edges by the total number of pairs in the system. The dark grey region of each box corresponds to the region between the first and second quartiles, the light grey corresponds to the second to third quartiles. The whiskers correspond to the pairs that are higher than the third quartile, and less than 1.5 times the interquartile range.The median probability for H pairs is around 1.8%, while the median probability for L pairs is around 33%.

significant step towards understanding better why some agents get matched and why some agents don't. In Appendix A.1 we compare the empirical distribution of edges and cycles to distribution obtained under a homogeneous and a 2-type Erdös-Renyi random graph.

Departures

Next, we focus on the departures. If a pair receives a match, it will leave the pool. The data shows, however, that many pairs depart without being matched. Among the 667 pairs included in our study 209 left the pool unmatched (124 were still waiting in the pool as of 07/01/2014, and 334 were matched). There could be a few reasons for such departures: the pair finds a match in another pool or from a deceased donor; the patient becomes too sick to transplant; the donor becomes unavailable. Abstracting away from the reasons for departure, in the following, we propose a probabilistic model for departures. [8] First in Figure 4-3 we

[8] We do not have full information on the departure reason.

illustrate the positive correlation between the PRA and average length of stay in the pool. Note that for patients with PRA of between 90 and 100, the median is 322 days, whereas for patients with a PRA between 0 and 10, the median stay is 7 days.

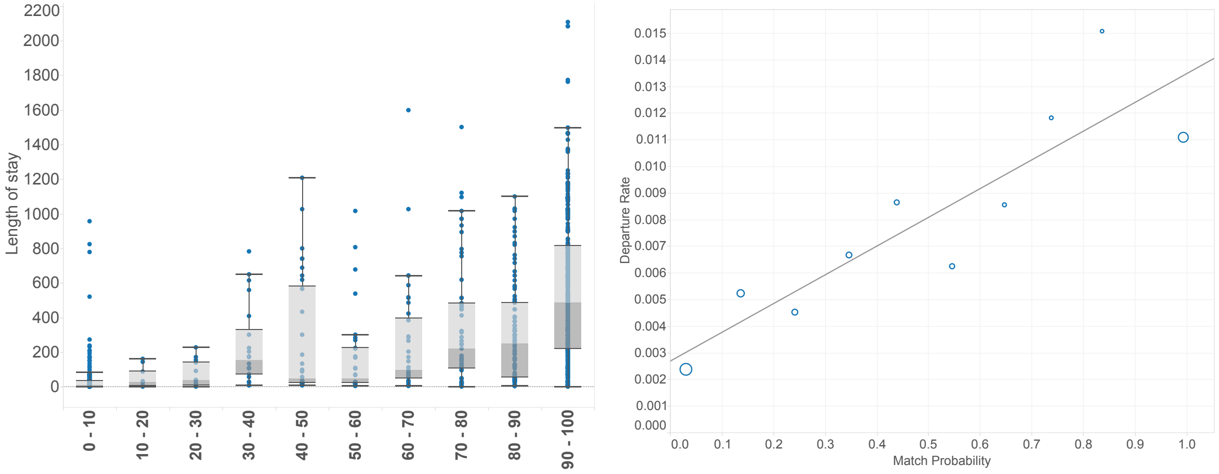

Figure 4-3: **Left:** Length of stay for non-transplanted patients as a function of their PRA. The box corresponds to the first and last quartile. The separation between dark and light gray is the median. **Right:** The Match Probability is calculated as $1-PRA$. it represents the probability to match with a blood-type compatible donor chosen at random. The Departure Rate is computed as the inverse of the average stay of all pairs of a given match probability. The regressions give an R^2 of 78%.

In Figure 4-3, the right plot shows that there is an (almost) linear relationship between the departure rate and the match probability. We observe that pairs with higher PRA and under-demanded pairs stay in the pool much longer than non under-demanded pairs with low PRA. Despite their significantly higher waiting time, a large number of high PRA pairs leave the pool unmatched. The data shows that 47% of low PRA pairs get a match while only 27% of high PRA pairs are able to get a match. Given the bimodal distribution of PRAs and the biais introduced by under-demanded pairs, we partition the pairs into 2 groups of H (high PRA) and L (low PRA) pairs, and fit 2 separate exponential distributions to the length of stay of pairs that left the pool unmatched. [9]

Figure 4-4 shows that the distribution of the length of stay for H and L agents condi-

[9] We elected to leave out the pairs that obtained a match in order to make this departure estimation independent of the matching process. This is a conservative estimation of the departure rate: being matched is likely to positively correlate with the ability to wait longer.

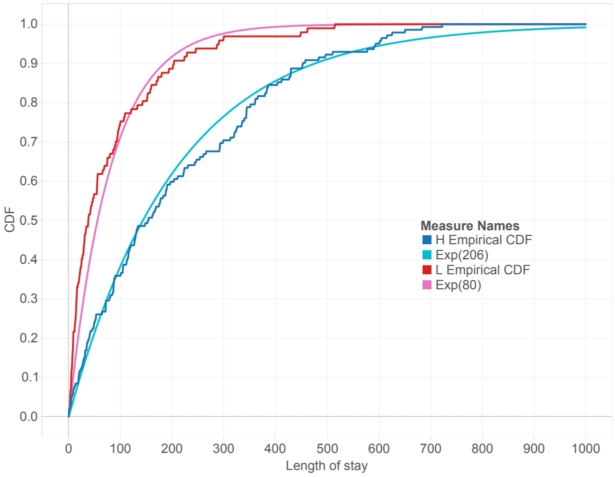

Figure 4-4: Empirical CDF for the length of stay of unmatched agents for H (blue) and L (red) pairs respectively, and the corresponding exponential best fits.

tional on not being matched can be estimated fairly well by an exponential distribution with respective means 206 and 80. The Lilliefors tests for goodness of fit are 0.027 and 0.038 respectively. Converting this into a discrete-time model of departures, we model the departure as follows: every iteration, a pair H (L) leaves the pool independently with probability $q_H \approx \frac{1}{1.28*206} \approx 0.0038$ ($q_L \approx \frac{1}{1.28*80} \approx 0.0098$), where we normalized to bring arrival rates to 1 per iteration Note that this implies that the length of stay of each pair H (L) in the pool follows a Geometric distribution.

Compatibility Graph

At any time, the set of incompatible pairs in the pool, V, induces a **compatibility graph** where a directed edge from v_1 to v_2 exists if and only if the donor of pair v_1 is compatible with the patient of pair v_2.[10] For each pair, we count the incoming edges in the following way: consider node v_2 that enters the pool at time T_1 and leaves at time T_2, for any pair v_1 that is present in the pool at some time between T_1 and T_2 and forms an edge to v_2, we count (v_1, v_2) as an incoming edge. [11] The number of such incoming edges gives an upper-bound

[10] In practice a minority of patients enroll with multiple donors. One can extend the model appropriately to capture this multiplicity.

[11] Note that we do not require v_1 to be in the pool during the whole horizon that v_2 stays.

on the matching opportunities that node v_2 had.

Figures 4-5 and 4-6 show the incoming degree distribution of H and L pairs that left the pool unmatched. We make the following crucial observation: **Even though H pairs stay in the pool much longer, they still have significantly less incoming edges (or equivalently matching opportunities) than L pairs.**

We observe that this incoming degree distribution has two components: a Binomial component corresponding to the incoming edges that form upon arrival, and a Geometric component corresponding to the edges that form from subsequent arrivals. For L pairs, the Binomial component dominates, in part due to their short stay in the pool. For H pairs however, the Geometric component dominates.

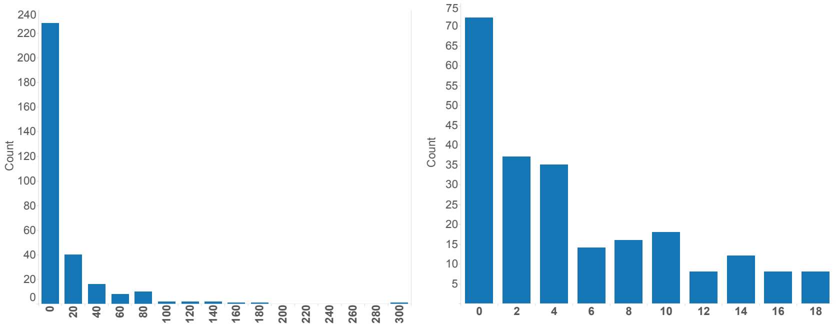

Figure 4-5: Distribution of the total number of ingoing edges an H agent observes during her lifetime in the system (both from pairs that were waiting upon its arrival, and from pairs that joined the pool subsequently). Left figure shows total histogram, right figure is the zoomed-in on the agents who see less than 20 edges during their stay in the system.

This raises the following question: Given the dynamics of kidney exchange pools, how can we improve the likelihood that a high-PRA patient gets matched? In the rest of the paper, we develop a dynamic model of kidney exchange that incorporate our key empirical findings, and we use theoretical analysis as well as data-driven simulation to compare different matching policies that determine when to match and how to prioritize.

In kidney exchange, matches are formed through both k-way cycles and chains (a chain is a directed path starting from an altruistic donor.). In this paper, we will focus on short cycles. [12]. Any improvement on policy for matching through cycles is complementary to the

[12] In practice, cycles of size either 2 or 3 are considered due to logistic reasons.

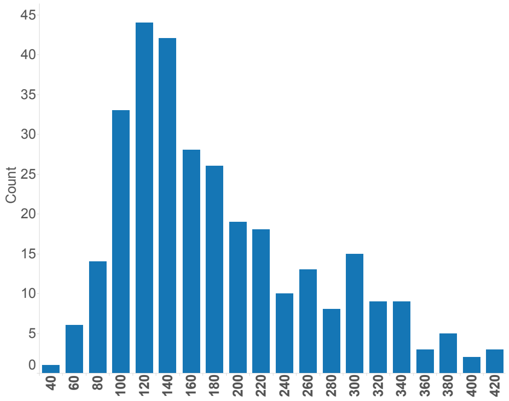

Figure 4-6: Distribution of the total number of ingoing edges an L agent sees during her lifetime in the system.

benefit of chains. Further, success of chains crucially depends on the existence of altruistic donors.

4.3 A data-driven model of kidney exchange

4.3.1 A dynamic model for kidney exchange

Motivated by our empirical findings, in this section, we present a stylized model that we will use in our theoretical study. Our model makes a few simplifying assumptions needed for tractability of analysis. However in our numerical study, presented In Section 4.5, we will closely follow all the empirical observation made in the previous section.

We study a discrete-time infinite-horizon model for dynamic kidney exchange that works as follows: At every time-step $t = 1, 2, \ldots$, one patient/donor incompatible pair arrives at the pool. Each pair corresponds to a node in the graph. Each node is one of two types, H (hard-to-match) or E (easy-to-match) thus modeling our bimodal PRA distribution. An arriving node is of type H independently and with probability $0 \leq \rho \leq 1$.

If the arriving node is of type H, it will obtain an incoming directed edge from any of the existing E nodes independently with probability p_H. However, it does not receive any incoming edges from another H node. In other words, H nodes cannot give to each other.

On the other hand, if the arriving node is of type E, it will form an incoming directed edge with any of the existing H (E) nodes independently with probability p_E (1). The outgoing edges are formed similarly. As data suggest (Figure 4-2) we will study the behaviour of the system in the limit when $p_H \to 0$, while p_E stays constant.

We will study matching using cycles. A k-way allocation or a k-way matching is a set of disjoint cycles each of size *at most* k. We focus on $k = 2$ and $k = 3$. [13] Based on the edge probabilities defined above possible 2-way cycles are $H - E$ and $L - L$, and possible 3-way cycles are $H - E - E$ and $E - E - E$. Figure 4-7 summarizes the edge formation and possible cycles.

Given that nodes depart continually, one natural policy is to try to find matchings every time a new pair arrives. However, one might also argue that by making pairs wait for *a short period of time*, we may increase matching opportunities especially for H pairs without increasing the departures. Our main theoretical result compares the performance of myopic (no-wait) policy to that of a policy that matches less frequently. In particular we focus on the following two policies:

Greedy: at every time step, finds a k-way allocation breaking ties in the following way:

(i) A 2-way cycle $H - E$ has the highest priority.

(ii) If $k = 3$, then $H - E - E$ has the second highest priority (if $k = 2$ skip this).

(iii) Finally an $L - L$ cycle has the least priority. Note by the greedy nature of this policy, nodes remaining in the pool (after the matching decision is made) do not form any cycles among themselves.

***b*-Batching:** at time steps $b, 2b, \ldots$:

(i) Find the maximum number of $H - E$ 2-way cycles, and remove them.

(ii) If $k = 3$, then find the maximum number of $H - E - E$, and remove them. (if $k = 2$ skip this).

[13] As mentioned before, in practice, cycles of size at most 3 are considered due to incentive and logistic reasons.

(iii) Match the remaining E nodes using $L - L$ or $E - E - E$ (if $k = 3$). Again, nodes remaining in the pool (after the matching decision is made) do not form any cycles among themselves.

The two policies are summarized in Algorithm 1.

Algorithm 1 k-way b-Batching ($b = 1$: Greedy)

```
For $t = b, 2b, \ldots$:
  1: Find maximum number of $H - E$ 2-way cycles, breaking ties randomly; remove matched nodes.
  2: If $k = 3$, find maximum number of $H - E - E$ cycles, breaking ties randomly; remove matched nodes.
  3: Match the remaining $E$ nodes using $E - E$ or $E - E - E$ (if $k = 3$), breaking ties randomly; remove matched nodes.
End for
```

Note that in both policies, there can be at most one E node left unmatched after looking for cycles. Following the rules defined above, it is easy to verify that *greedy* policy is equivalent to *b-batching* with $b = 1$. To distinguish these two policies, in the rest of the paper we assume $b > 1$.

Next we present our simplified model of departures: we assume nodes only depart at the beginning of time steps $b + 1, 2b + 1, \ldots$ (or the beginning of a batch). At time $\tau b + 1$, if the node that arrived at time $t \leq \tau b$ is of type H and is still in the pool, it departs with probability bq_H. However, if such a node is of type E, it will depart with probability 1. [14] Finally motivated by Figure 4-3, we assume $q_H = \alpha p_H$ where $\alpha < 1$. The sequence of events is assumed to be the following: at each time step t:

1. Departure: if $t = \tau b + 1$ for some integer τ, then departing nodes are removed.

2. Arrival: A pair arrives.

3. Matching: In greedy policy the match is decided at the end of every time step t. In batching-b a match is decided only if $t = \tau b$.

[14] This is not a restrictive assumption in our model because for both of the policies defined above, at the end of a batch, we can have at most one E vertex left.

See section 4.4.1 for a more in-depth discussion of our modeling assumptions.

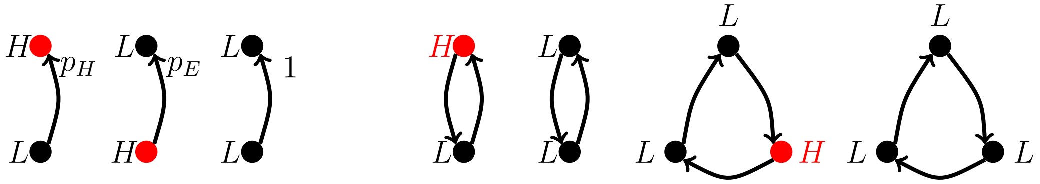

Figure 4-7: Left: Directed edge formation; Right: Possible matching cycles

4.3.2 Preliminaries

We denote the number of H nodes at the beginning of the τ-th batch as Π_τ^b and Π_τ^G for the *b-batching* and the *greedy* policy respectively. Dynamics described above imply that Π_τ^b and Π_τ^G are both Markov chains: arrivals are i.i.d; departures follow a memoryless distribution. Further, at the beginning of a batch the pool consist of Π_τ H nodes and at most one E node which will immediately depart. Also, excluding that possible E node, all the nodes in the pool have in-degree/out-degree of zero, and all the future edge formations are also i.i.d and independent of the index of a node. Because b is fixed number, The Makrov chains Π_τ^b and Π_τ^G have negative drifts when Π_τ^b and Π_τ^G become "too large": in expectation, $bq_H\Pi_\tau$ pairs leave the pool at the beginning of the batch. This property ensures that the chain reaches steady state:

Proposition 1. *Makov chains Π_τ^b and Π_τ^G reach steady state.*

Let $\mathbb{E}\left[\Pi_\infty^b\right]$ and $\mathbb{E}\left[\Pi_\infty^G\right]$ be the expected number of H nodes in steady-state. Further, let $\mu_k^b(\tau)$ and $\mu_k^G(\tau)$ denote the number of H nodes matched during the τ-th batch when only cycles of length up to k ($k \in \{2, 3\}$) are allowed. With a slight abuse of notation, we will denote by $\mathbb{E}\left[\mu_k^b(\infty)\right]$ and $\mathbb{E}\left[\mu_k^G(\infty)\right]$ the expected number of H nodes matched in one batch in steady-state. Balancing the arrival and departure in steady-state, implies that: [15]

[15] Note the pool size also depends on k, but to keep the notation simple, we use Π_τ^b (Π_τ^G) regardless of value of k.

$$b\rho = \mathbb{E}\left[\mu_k^b(\infty)\right] + bq_H\mathbb{E}\left[\Pi_\infty^b\right] \qquad b\rho = \mathbb{E}\left[\mu_k^G(\infty)\right] + bq_H\mathbb{E}\left[\Pi_\infty^G\right] \tag{4.1}$$

The immediate implication of (4.1) is that $\mathbb{E}\left[\Pi_\infty^b\right] \leq \rho/q_H$ and $\mathbb{E}\left[\Pi_\infty^G\right] \leq \rho/q_H$. Later in Lemma 10 we prove that the same upper-bound holds "with high probability" for Π_τ^b and Π_τ^G. Further, in Lemmas 11 and 12 we show a lower-bound on both Π_τ^b and Π_τ^G of the form $l\rho/q_H$ for a constant $l \in (0, 0.5)$. These bounds show that the expected size of the pool is controlled by q_H (or equivalently p_H). The smaller p_H is, the larger the pool becomes.

Before proceeding with our analysis, we compute the in-degree distribution of the two types to ensure our model follows the trends we see in data. For a moment suppose we do not batch the departures together. Suppose each H node departs at the beginning of each step with probability q_H. Consider an H node v that departs from the pool unmatched. The number of incoming edges that node v forms after arrival is Geometric variable with mean $\frac{(1-\rho)p_H(1-q_H)+q_H}{q_H} = \frac{(1-\alpha p_H)+\alpha}{\alpha}$[16]: this follows form the the thinning property of the Geometric distribution.

Now let us compute the number of incoming edges that node v sees upon arrival. If the greedy policy is being exercised upon arrival of v, then at most one E node is in the pool. Therefore node v will form one edge at most with probability p_H. If b-batching is being used, there can be at most $B \sim Binomial(b, (1-\rho))$ E nodes in the pool, therefore node v forms at most $Y \sim Binomial(B, p_H)$ incoming edges which is still assymptotically small in p_H when b is a constant. This highlights the following point: *for small values of b, b-Batching does not have a significant impact in increasing in-degree of H nodes.* However, batching still can be beneficial because it can lead to better matching decisions. In the next section we show that when using only 2-ways such a gain remains limited and diminishes for p_H small enough. However, when we match also through 3-ways, b-batching leads to an increase in matching rate that does not scale down with p_H. Overall, putting the calculation for in-coming edges upon arrival and during the stay, we see that even in the limit of $p_H \to 0$ the in-degree of H

[16] Recall that we assume $q_H = \alpha p_H$.

nodes remains a small constant. This capture the sparsity of the exchange pools.

Finally, let us also study the in-degree of E nodes. In steady-state, using the lower bound above, with high probability, there are at least ${}^{l\rho}/_{q_H}$ H nodes present. Therefore, upon arrival of an E node, it receives at least ${}^{l\rho p_E}/_{q_H} = {}^{l\rho p_E}/_{\alpha p_H}$ incoming edges Given that $p_H \ll q_E$, it can already be seen that E nodes form many more incoming edges even upon arrival.

4.4 Main theoretical results

In this section we present our theoretical analysis which focuses on comparing the greedy and the b-batching policy. We will use the expected match rate of H nodes in the steady-state (i.e., $\mathbb{E}\left[\mu_k^b(\infty)\right]$ and $\mathbb{E}\left[\mu_k^G(\infty)\right]$)) as the measure of comparison.[17] The larger the match rate, the smaller the pool size.[18] Further, it follows from the Little's law that the larger the match rate the smaller expected wait-time in the steady-state.

In our first theorem, we focus on $k = 2$, and show that batching does not improve significantly the number of matches over the greedy policy. More precisely, we show the gain from b-batching scales at most linearly with p_H. As we shall see in Theorem 2, this is not the case when 3-way cycles are allowed.

Theorem 1. *Suppose $k = 2$, in steady-state we have:*

$$\mathbb{E}\left[\mu_2^b(\infty)\right] \leq \mathbb{E}\left[\mu_2^G(\infty)\right] + p_H\lambda$$

where $\lambda = (1-\rho)p_E b^2\left[\rho/2 + (1-bq_H)p_E\rho/\alpha + p_H p_E b - p_H p_E(1-\rho)^2 b + 2\right]$.

Remark 1. *In the limit $p_H \to 0$, $\lambda \to (1-\rho)p_E b^2\left[\rho/2 + p_E\rho/\alpha + 2\right]$, and therefore, $p_H\lambda \to 0$.*

The detailed proof of this theorem is presented in Appendix A.2. Here we only describe the high level ideas of the proof.

[17] As (4.1) shows, larger match rate is equivalent to the smaller loss rate: the expected loss is proportional to the expected number of H nodes in pool.

[18] Note that we use "pool size" and number of H nodes in the pool interchangeably .

Proof idea of Theorem 1. In the proof, we focus on analyzing the pool sizes associated with the the policies (i.e., Π_τ^b and Π_τ^G). Then we use (4.1) that relates the pool size with the match rate. In order to prove the result, we show in steady-state $\mathbb{E}\left[\Pi_\infty^G\right]$ is not "much larger" than $\mathbb{E}\left[\Pi_\infty^b\right]$.

We show that for the same pool size, the gain of applying the b-batching instead of greedy at any step is asymptotically small in p_H; This implies that the gap between Π_τ^b and Π_τ^G remains "small". The intuition behind the proof is as follows: after a new batch arrives, let us look at the undirected graph formed between the new batch and all the H nodes in the pool at the beginning of the batch (remember that all the nodes in the pool are isolated, meaning have in-degree/out-degree zero). Excluding the E-E edges, this graph is is extremely sparse and disconnected (See Figure 4-8). The maximum matching in a disconnected graph is the union of the maximum matching of each of its connected components, and for each of these components, b-batching cannot do any better than greedy. For instance consider the example of Figure 4-8; when r_1 arrives, it forms its three undirected edges. Most likely, nodes c_1, c_2, and c_3 will not have any other neighbors in this batch (the filled nodes in Figure 4-8), and thus the decision of greedy and b-batching would be equivalent. In the proof we formalize this intuition by upper-bounding the size of the augmenting path to the matching obtained the greedy policy. The rest of the proof deals with analyzing the evolution of the gap between the two pool sizes.

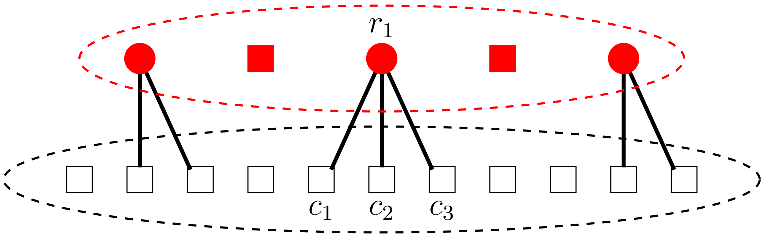

Figure 4-8: The typical connected components (excluding the E-E edges) in the b-batching; the circle nodes are of type E and the rectangle ones are of type H; the filled nodes are the ones which arrived in the last batch, and the not-filled nodes are those which arrived previously and have not left the pool yet.

□

Next, we study the case where cycles of length 3 are alllowed, and compare the matching rates of the *b-batching* and *greedy* in steady-state (i.e., $\mathbb{E}\left[\mu_3^b(\infty)\right]$ and $\mathbb{E}\left[\mu_3^G(\infty)\right]$). In particular, we state the counterpart of the above theorem when 3-way cycles are allowed, and show that unlike the 2-way matching (where waiting is not effective), there are some cases (depending on p_H, p_E, ρ and α) where b-batching can match many more H nodes than greedy does, and the gain of batching does not scales down with p_H.

Theorem 2. *Suppose $k = 3$, $\epsilon = \max\left\{\sqrt{\frac{6q_H}{0.5\rho}}, \sqrt{\frac{4q_H[1+(1-\rho)/\alpha]}{0.5\rho(1-p_H p_E)}}\right\}$, and parameters ρ, p_H, q_H, and p_E are such that $\epsilon < 1$. In steady-state we have:*

$$\mathbb{E}\left[\mu_3^b(\infty)\right] \geq \mathbb{E}\left[\mu_3^G(\infty)\right] - \gamma p_H + \bar{\gamma} p_E^2$$

where $\gamma = \frac{2b^2(1-\rho)(1-bq_H)}{1+(1-\rho)(1-bq_H)/\alpha}$ and

$\bar{\gamma} = \frac{\frac{1}{2}(1-bq_H)^2(1-p_H p_E)^{2\rho b}(1-p_H)^{2\rho b}\left(\frac{1-p_E}{2(1-p_H)}\right)^2(0.72\xi)^2}{1+(1-\rho)(1-bq_H)/\alpha}.$

for ξ the solution of $\min_{x\in[2(1-\epsilon)\frac{0.5(1-p_H p_E)}{1+(1-\rho)/\alpha}\rho/\alpha, 2(1+\epsilon)\rho/\alpha]} xe^{-x}$.

Remark 2. *In the limit $p_H \to 0$,*

(a) If $\rho > 0$ then $\epsilon \to 0$.

(b) $\gamma \to \frac{2Z^2(1-\rho)}{1+(1-\rho)/\alpha}$, therefore $\gamma p_H \to 0$.

(c) $\bar{\gamma} \to \frac{\frac{1}{2}\left(\frac{1-p_E}{2}\right)^2\left[0.72\min_{x\in[2\frac{0.5}{1+(1-\rho)/\alpha}\rho/\alpha, 2\rho/\alpha]} xe^{-x}\right]^2}{1+(1-\rho)/\alpha}$, therefore

$$\bar{\gamma} p_E^2 \to \frac{\frac{1}{2}\left(\frac{1-p_E}{2}\right)^2\left[0.72\min_{x\in[2\frac{0.5}{1+(1-\rho)/\alpha}\rho/\alpha, 2\rho/\alpha]} xe^{-x}\right]^2 p_E^2}{1+(1-\rho)/\alpha}.$$

The detailed proof of this theorem is presented in Appendix A.3. Here we only outline some of the proof ideas.

Proof idea of Theorem 2. Similar to the proof of Theorem 1, we focus on analyzing the pool sizes associated with the the policies (i.e., Π_τ^b and Π_τ^G). Then we use (4.1) that relates the pool size with the match rate. Here we show in steady-state $\mathbb{E}\left[\Pi^G_\infty\right]$ is "significantly larger" than $\mathbb{E}\left[\Pi^b_\infty\right]$. The gap stems from the following observation: during the arrival of a batch, it can happen that an arriving E node v forms a directed edge to an (existing) H-node u, but does not receive back an edge from u. Under greedy policy, it is likely that node v forms an $E-E$ cycle immediately or shortly after its arrival, and leave. This would be a missed opportunity for matching node u which only sees a few edges in its entire stay in the pool. On the other hand, *b-batching* is likely to make a different decision: because it waits until the end of the batch, with some probability it will find another E node arriving in the same batch to form an $H-E-E$ 3-way which matches node u. Therefore, b-batching matches more H nodes by "wasting" fewer E nodes, which makes the pool associated with the greedy larger. A major part of the proof deals with finding a lower bound on the number of such $H-E-E$ 3-cycles that greedy "misses" but b-batching does not. What makes the counting particularly challenging is the non-monotone relationship between the pool size and probability of the aforementioned event: if the pool is larger, the chance of a directed edge from E to H increases (which is the necessary part of forming the beneficial $H-E-E$ cycle), but at the same time the probability that the same E node forms a $H-E$ cycle immediately also increases. These factors contribute into the rather cumbersome form of the constant $\bar{\gamma}$.

The second part of the proof resolves the following dilemma: the pool associated with greedy becomes larger over time due to the loss in matching as explained above; The match size in each batch increases as the pool becomes larger. Can this increase in match size (as a result of a larger pool) create a large enough drift to bring back the size of the pool for greedy close to the one for b-batching? We show that in a pool with X extra nodes, the expected increase in match size only scales with $p_H X$, and therefore for p_H small, the increase is "insignificant". The factor $-\gamma p_H$ in the theorem statement accounts for this drift that goes in the opposite direction of the gain from batching.

□

4.4.1 Discussion on model and theoretical results

The above theoretical results are founded on a few fundamental properties of the underlying compatibility graph. These properties qualitatively hold beyond the simplified model we used to keep the analysis intractable. Here we use the insight from our analysis to show how the results will change if we relax some of the modeling assumptions.

One of the main simplifying assumptions we made was that nodes do not depart during a batch. This is indeed a strong assumption that we need to make only because of another strong (and unnecessary assumption) in b-batching policy which requires *all* pairs in the arriving batch to wait until the end of the batch. As our analysis shows if at any time we can match an H node, there is no significant benefit from making this node wait (the example in Figure 4-8 is one illustration of this insight). In fact, the only scenario for which waiting helps is in 3-way matching when an E node can give to an H, but cannot form a cycle right away.

Therefore, in the model where nodes depart at every time-step, we can modify the b-batching to an *opportunistic b-batching* policy which works as follows: if $k = 3$, during the batch we only make an E node v wait if v forms an edge to at least one H node (say node u) without forming any cycle involving an H node. At any later time, if another E node arrives that does not form any $H - E$ cycle but forms a 3-way with u and v, we will execute the match right away. We let the rest of the nodes match immediately.[19]

Comparing the opportunistic b-batching with the greedy policy (under the relaxed assumption that nodes depart every step), we expect the gain from making these special E nodes wait to remain significant: following the lines of proof for Theorem 2, we do not use the fact the nodes other than these special E nodes also wait until the end of the batch. Further, our data driven simulation study also confirms that, for $k = 3$, such opportunistic batching results in a significant increase in match rate of H nodes compared to that of the greedy policy (see Figure 4-11).[20]

[19] For $k = 2$, Theorem 1 already shows there is not a significant benefit from waiting, so we will not use b-batching.

[20] A theoretical analysis of *opportunistic b-batching* is quite challenging because it would require us to

Next we note that the results are stated in a way to highlight the dependency of each gain/loss term on p_H which is a very small number. As Remarks 1 and 2 illustrate, the asymptotic case of $p_H \to 0$ offers the clearest dichotomy between the comparison for 2-way matching (where the two policy will have the same matching rate) and 3-way matching (where the b-batching has a non-vanishing gain). For a fixed value of p_H, our comparison of the match rates involve constants λ, γ, $\bar{\gamma}$. The combinatorial and stochastic nature of the matching process makes it extremely difficult to optimize these constants or compute the exact matching rates. However, we use data-driven simulation (presented in Section 4.5) to confirm that our assertion based on our theoretical analysis also holds in settings driven by real data.

Finally, assuming that E nodes match each other with probability 1 allowed us to merely focus on match rate for H nodes which is indeed the main challenge in kidney exchange. However, as discussed in Section 4.5, there is a tradeoff between matching H nodes and E nodes: when we make some E nodes wait (to help the H ones), we increase the chance that they leave the pool unmatched. This highlights the limit of batching policy in increasing the total match rate (including both types).

4.5 Data-driven simulation results

In this section we provide simulations that implement the different matching policies that we considered in Section 4.3. We use the compatibility graph obtained in the clinical data from NKR. In addition, because compatibility can deterministically be determined using antigens, antibody levels and blood-types, we were able to know whether two pairs that may have never been in the NKR pool at the same time would have been compatible. We simulated arrivals and departures according to the models derived in Section 4.2.

analyze a high dimensional Markov chain that not only keeps track of number of H and E nodes, but also the existing edges among them.

4.5.1 Simulation set-up

In order to model the departure process, we fit an exponential distribution on the length of stay for agents that were not eventually matched. We controlled for hardness-to-match by fitting the departure distribution independently for H and E pairs.

Because we do not theoretically model blood types, we removed U agents from our simulations. This can be thought of as the matching process when we focus on the sub-pool of blood-type compatible pairs. We show simulation results in a more general case in section 4.5.4.

In our simulation, we normalized the arrival rates to 1 per time step Therefore, we normalized the departure rates to $q_H = 0.0038$ and $q_E = 0.0097$, as discussed in section 4.2. At each iteration, we sample with replacement one pair from the pool of 1677 pairs that arrived to the NKR between 01/2008 and 07/2013. [21]

In order to provide a simple yet flexible framework, we implemented integer formulations for the weighted matching problem in settings where either 2-cycles or both 2 and 3-cycles are allowed. We then ran these formulations every b iterations where b is the batch size.

4.5.2 Implementation of our theoretical policies

Recall that there can be multiple reasons for a pair departing: it can obtain a match in a competing exchange program, or get a kidney from the deceased waiting list, or the patient can become too sick to become transplanted.

Fielded Kidney Exchange programs such as NKR have began implementing such priorities in some cases: pediatric matches are given a higher priority for example, perfect matches [22] also get a higher priority. In Section 4.5.3 we consider the following policies:

- **No-priority(b):** this is the policy where we solve a maximum-cardinality matching

[21] While it was important to consider only recent data to get a good estimate of the arrival and departure rates, we chose to take all the pairs into account to increase diversity and limit the biais introduced by sampling the same agents multiple times

[22] Perfect match occurs when the donor's antigenes and the patient's antibodies are perfectly compatible. They reduce the likelihood of a positive crossmatch, and ultimately the benefit of the transplant to the patient.

problem, without prioritizing hard-to-match vertices. This means that all pair types are treated the same and ties between two options that match the same number of pairs are broken randomly.

- **Batching(b):** this is the policy that is closest to our theoretical analysis. E pairs are given a weight of 2, and H pairs are given a weight of 5. In the case of $b = 1$, this is the greedy policy, and there is never more than a single cycle matched at any time step. This enables the following hierarchy of priorities: $H - H - H > H - H - E > H - H > H - E - E > H - E > E - E$.

We note that there are two potential issues with this prioritizing scheme. Firstly, even though $E - E$ has the lowest priority, it leads to a loss of E pairs that could potentially help future pairs match. Secondly, prioritizing $H - E - E$ over $H - E$ can have similar results. This leads us to define a new matching policy:

- **Opportunistic Batching(Z):** This policy reduces the loss of H pairs due to batching by matching them as soon as possible. It assigns a weight of -2 to E pairs that have been in the system less than b iterations and have at least one edge that points to an H agent in the system. E pairs that have been in the system longer get a weight of $+1$, and H pairs get a weight of 5. This prevents $E - E$ matching for the first b time steps that they are in the system, but allows for $E - E - H$ as well as $E - H$.

4.5.3 Results

In order to reach the steady-state distribution, we ran each simulation for 10,000 iterations, and all the statistics that we show are averaged over iterations 2,000 through 10,000 (note that transient state lasts around 500 iterations, depending on the matching policy employed). Because there are still variations between runs, we ran 20 independent simulations for each setting in order to make sure that our findings are statistically significant. In Figures 4-9 through 4-13, we plot the match rates obtained in all 20 runs, for the different pair types (all, only H, only E, etc) for different settings. We also show box-plots: the region between the first quartile and the median is shown in dark grey, the region between the median and the third quartile is shown in light grey. The whiskers extend to the closest point that is distant from the box by at most 1.5 times the inter-quartile range.

No-priority

In this section, we consider the effects of batching under the no-priority policy, for batch sizes varying between 1 and 50. In Figure 4-9 we observe that even though H agents stay almost three times longer in the system, their match rate is significantly lower than that of the E agents.

We also observe that as the batch size increases, the match rate for E pairs decreases, while it increases for H pairs. This can be explained by the fact that H agents have a lower departure rate, and therefore batching them for a few time steps will not harm them much. On the other hand, because of the batching mechanism, some H agents are able to match to E pairs that would otherwise have matched together. While the overall match rate decreases, this illustrates the fact that when matching H agents is deemed more important than matching E agents, some level of batching may be useful.

Prioritizing

In this section, we analyze how the batching policy (which breaks ties in favor of H agents) performs. As shown in Figure 4-10, compared to the no-priority policy, batching increases the median match rate of H pairs by 6 percentage points, while decreasing the median match rate of E pairs by 8 percentage points. Overall, there is no significant difference in the total match percentage. We notice that this improvement for H pairs is much stronger than the one observed when batching with the Greedy policy.

Furthermore, increasing the batch size does not seem to yield a significant improvement for the prioritizing policy in terms of the number of H pairs that are matched. There are two competing forces here: on one side, increasing the batch size gives H pairs more E pairs with whom to match. On the other side, the greater the batch size, the more likely it is that

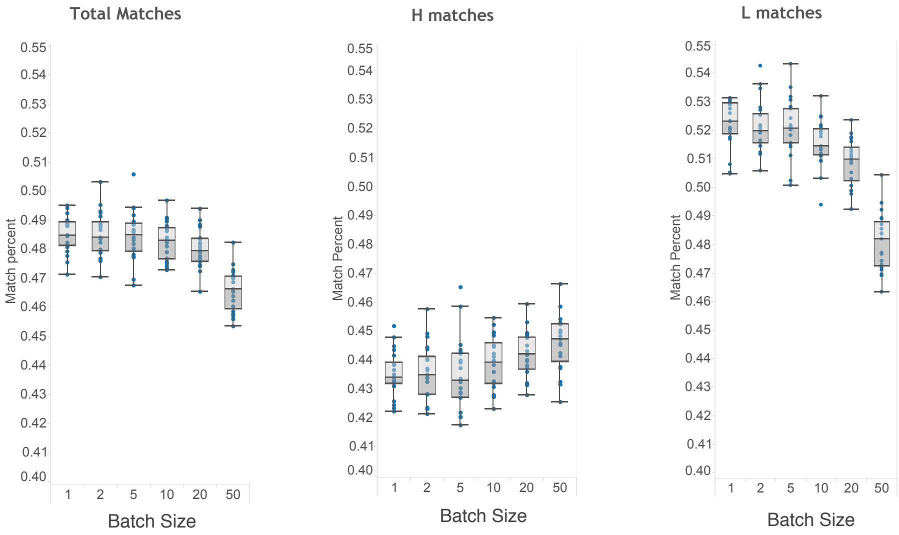

Figure 4-9: Comparison of match rates under the **No-priority policy**, for varying values of the batch size (b = 1,2,5,10,20,50), when both 2 and 3-cycles are allowed. The left graph shows the aggregate match rate of H and E. The middle graph shows the match rate for H pairs, and the right graph shows the match rate for E pairs.

an H pair will depart when we could have found a match.

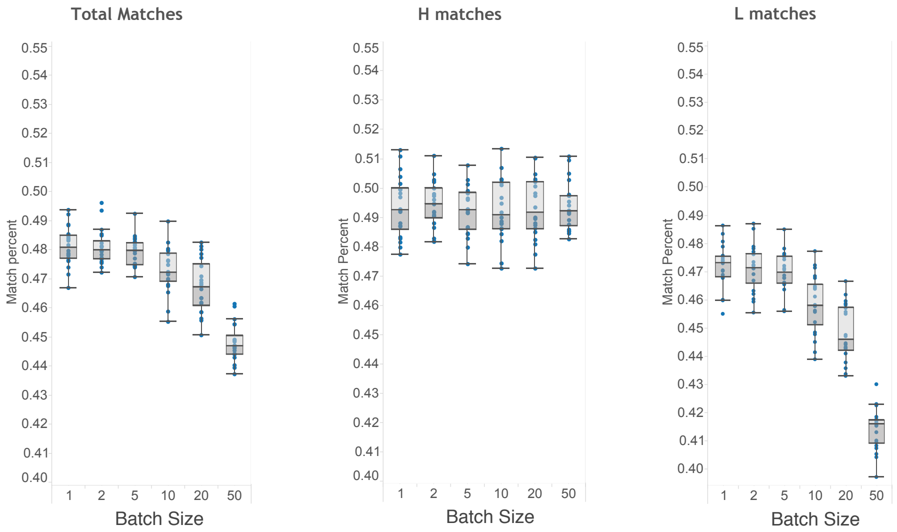

Figure 4-10: Comparison of match rates for the **Batching policy** for batch sizes of between 1 and 50, when both 2 and 3-cycles are allowed. The left graph shows the aggregate match rate, the middle graph shows the match rate for H pairs, and the right graph shows the match rate for E pairs.

Opportunistic Batching

One question we want to ask now is the following: can we improve H pairs' match rate beyond what we get with prioritizing? To answer this, we look at our Opportunistic Batching policy. Figure 4-11 shows that we can indeed slightly improve the median matching rate of H agents from 49% to 51%. However, this reduces the median matching rate for E pairs by 7 percentage points, from 48% to only 41%. We leave it to the policy maker to decide whether such a tradeoff is acceptable.

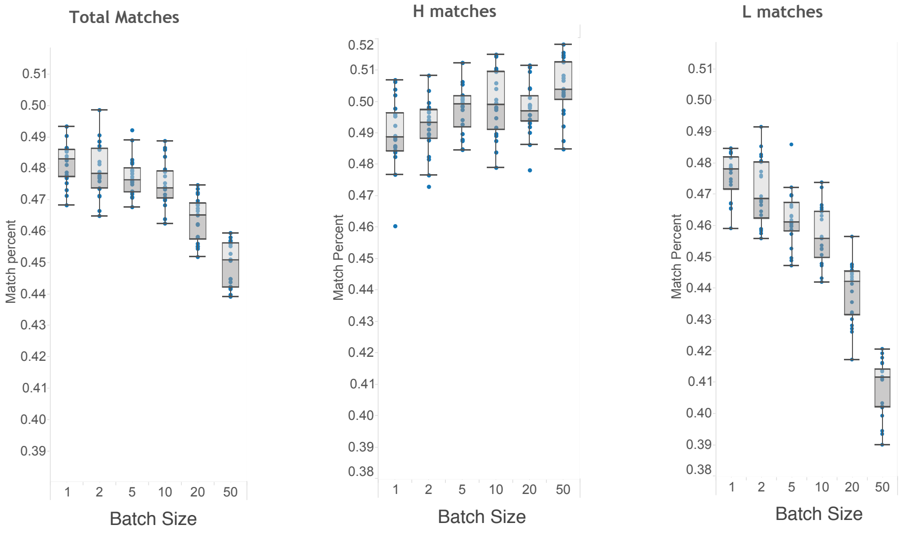

Figure 4-11: Match rates for the **Opportunistic batching policy**, and batch sizes of 1, 20 and 50 when cycles of length 2 and 3 are allowed. Left graph is the overall match rate. Middle graph is the match rate for H pairs, and Right graph is the match rate for E pairs.

4.5.4 Extensions

Sensitivity analysis on arrival and departure rates

We observed in Section 4.5.3 that batching has a limited effect in today's NKR pool. An interesting question is to determine whether this is an intrinsic property of the system, or whether this is a consequence of the current arrival and departure rates. In other words, what would happen if we were to increase the arrival rate of new pairs?[23] In Figure 4-12 we

[23] Because we normalize arrivals, this is equivalent to decreasing the departure rates.

observe match rates under the batching policy when we double the arrival rate (or we divide the departure rate by two).

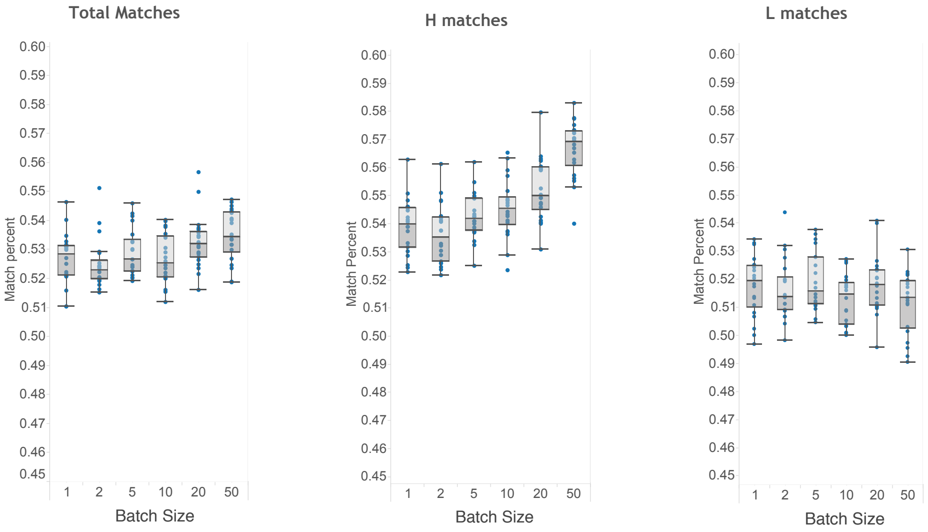

Figure 4-12: Match rates for the **Batching policy**, when both 2 and 3-cycles are allowed, in the case where the departure rates are divided by two: $q_H = 0.0019$, $q_E = 0.0049$. The left graph shows the aggregate match rate, the middle graph shows the match rate for H pairs, and the right graph shows the match rate for E pairs.

We observe that for an increased arrival rate, some level of batching leads to an increased match rate for H pairs, and has a limited effect on E pairs. We note that here, the batch size is measured in terms of the number of arrivals between match runs. In practice however, matches are run periodically (daily or weekly) irrespective of the number of arrivals in the interval. The optimal level of batching and how does it relate to the arrival and departure rates is an interesting open question.

Under-demanded pairs

In this section, we show simulations that go beyond the settings of our theoretical analysis. In particular, we observe what happens to the matching policies that we analyzed in settings with under-demanded agents. In addition to our previous two types H and E, we now introduce the U pairs, which correspond to pairs with under-demanded blood-types and low PRA (≤ 80).

First we introduce under-demanded agents to our simulations. This means that our arrivals are now sampled uniformly from the data in the NKR pool. We compute the estimated departure rates based on the average stay in the system for each type of pair: $q_H \approx \frac{1}{1{,}28*206} \approx 0.0038$, $q_E \approx \frac{1}{1.28*80} \approx 0.0097$, and $q_U \approx \frac{1}{1.28*177} \approx 0.0044$.

In Figure 4-13, we show how match rates vary, both for the greedy and the prioritizing policy, as a function of the batch size. We observe that similarly to the case without any under-demanded pair, batching only helps H pairs when there are no priorities, and prioritizing helps H agents more than batching.

However, we also see in Figure 4-13 that prioritizing for H pairs sharply decreases the match rate for U pairs. Somewhat surprisingly, batching using the greedy policy also seems to increase the match rate for U pairs. One explanation for this is that as the batch size increases, there are less $E - E$ matches which negatively impacts the match rate for under-demanded pairs as well as H pairs This would make it an interesting compromise between helping H agents and helping U pairs.

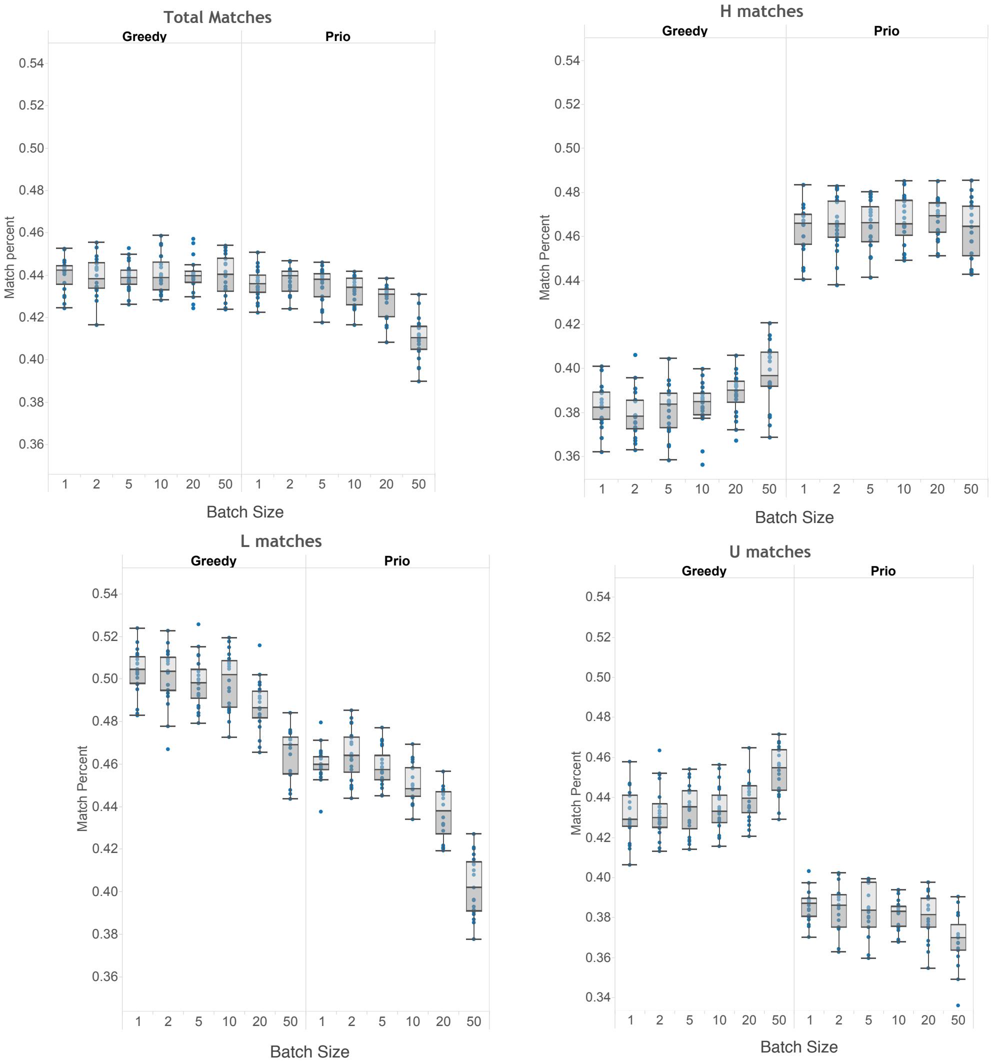

Figure 4-13: Comparison of match rates for greedy and prioritizing and varying batch sizes (1 to 50). Top right: overall match rate, top left: match rate for H pairs, bottom right : match rate for E pairs, bottom left: match rate for U pairs..

Chapter 5

On Matching and Thickness in Heterogeneous Dynamic Markets

5.1 Introduction

This paper is concerned with the problem of matching in a dynamic marketplace, where heterogeneous agents arrive over time to the market looking to exchange an indivisible item for another compatible item. A key feature of the market is its exogenous thickness, as determined by the types of agents and their arrival rates to the marketplace. For example, in kidney exchange some patient-donor pairs are very hard-to-match while others are very easy-to-match. In online labor markets, employers have different qualification requirements and workers have different skills.

Efficiency is determined by the matching policy and the matching technology. The former determines which exchanges to be implemented and when, and in particular which priorities, to assign to different types of agents. The latter determines the forms of matches that can take place; For example, while kidney exchanges were first conducted through bilateral exchanges (2-way cycles) [105], multi-hospital platforms are now facilitating many of their transplants through chains initiated by altruistic donors [7]. In many matching markets, such as dating, only bilateral matches take place.

We are interested in the behavior of simple *myopic policies* under different matching technologies and different thickness levels of the market. Myopic policies form matches as soon as they become available, but may vary with respect to how they prioritize agents in the events of ties. Our framework will allow to discuss policy questions such as: What is the effect of prioritizing different types of agents? How does disproportional change in arrival of different types influence market efficiency? What is the impact of merging matching marketplaces with different thickness levels on different types?

Two comments are in place. First, restricting attention to myopic policies is motivated by current practices in kidney exchange platforms in the United States. Ashlagi et al. [17] uses simulations based on empirical data from multiple exchange programs to show matching myopically is nearly harmless. Moreover, a similar conclusion is also arrived in theoretical work [8]. [1] While they consider a stylized model with homogeneous agents, their result can

[1] See Subsection 5.1.1 for further details.

be generalized to our heterogeneous model.[2]

Second, the literature on dynamic matching in sparse environments has focused on homogeneous agents [4, 8]. The motivation for this paper stems from the heterogeneity of agents in the marketplace.

For our purposes we propose a simple infinite-horizon model with two types of agents, easy-to-match (E) and hard-to-match (H). Agents of each type T arrive to the marketplace according to an independent Poisson process with rate λ_T. Each agent arrives with an indivisible item that she wishes to exchange. We assume a stochastic demand structure, where each agent of type T finds the item of any other agent compatible independently with probability p_T. A key feature of the model is that p_H is significantly smaller than p_E. Agents are indifferent between compatible items but prefer to be matched as early as possible. Moreover, agents in our model depart the market after being matched. We therefore adopt the average waiting time of agents in steady-state as a measure for efficiency.[3] While our model is highly stylized, it captures some important features observed in kidney exchange pools (see Section 1.2.1, where we provide a brief background that further motivates this study).

Two settings are considered, distinguished by how matchings are formed (feasible exchanges): bilateral (2-way cycles), and chains. Our main findings are the following. First, we find that market thickness plays a crucial role on the desired matching technology; when easy-to-match agents arrive more frequently to the market than hard-to-match ones, the average waiting time of H agents scales similarly under chains and bilateral matchings. But there is a sharp increase in the average waiting time of H agents as soon as hard-to-match agents arrive more frequently, highlighting the importance of chains in marketplaces with a majority of hard-to-match agents. Second, we find that, under bilateral matching, increasing arrival rates of hard-to-match agents may negatively affect hard-to-match agents by increasing their waiting times. Under chains, however, increasing arrival rates always

[2] This is not the focus of this paper, but for completeness, we show this in Appendix B.8.

[3] More precisely, we focus on the average waiting time of H agents, because the waiting time of E agents is negligible compared to that of H agents. For a more detailed discussion, see Section 5.2.

shortens waiting times. Third, impact of prioritization in bilateral matching also depends on the market composition; when hard-to-match agents are the minority type, assigning them priority improves their waiting times. [4] However, when they are in majority, such prioritization has no significant impact.

Next we describe our results more formally under the bilateral and chain settings. In our analysis we compute the average waiting time of H agents under various myopic policies as $p_H \to 0$.

Bilateral matching. Two myopic policies are considered for bilateral matching, differing in the type of agents they prioritize; While it appears natural to prioritize hard-to-match agents, it is also interesting to consider the prioritization of easy-to-match agents as these may have better outside options.[5]

We find that regardless of how agents are prioritized, when $\lambda_H < \lambda_E$ waiting time scales with $1/p_H$, and when $\lambda_H > \lambda_E$ waiting time scales with $1/p_H^2$. When easy-to-match agents arrive more frequently, prioritizing H agents results in shorter waiting times than prioritizing E agents. However, when $\lambda_H > \lambda_E$, the average waiting time in the limit is identical under both types of priorities.

We further provide comparative statics for the case in which H agents are prioritized. Increasing λ_E always decreases waiting times. However, the average waiting time is non-monotone when increasing λ_H; It has an increasing trend up to a certain threshold, which depends on λ_E, and then it decreases (note that in a homogeneous model with only H agents, Little's law implies that increasing λ_H always decreases waiting times). These findings have two main implications: (i) thickening the market by increasing arrival rates of hard-to-match agents can result in longer waiting times depending on the existing arrival rates, (ii) merging two marketplaces with different compositions, i.e. different ratios between the two arrival

[4] To be precise, theoretically we are only able to prove that prioritizing H agents leads to shorter or equal waiting times (of H agents), however, numerically we confirm that such prioritization indeed leads to strictly shorter waiting times.

[5] In reality, agents may leave the market without being matched due to various reasons such as finding outside options. Under a stochastic departure model, shorter waiting times correspond to fewer departures because both quantities are proportional with the market size. We leave the rigorous treatment of a model with departure as an open question.

rates, may not be beneficial for both.

Chain matching. Under the chain setting, we consider policies termed *ChainMatch(d)* for markets endowed with d altruistic donors who initiate chains that continue indefinitely. In a chain, each agent is matched by (receives an item from) some agent, and matches another. Whenever the last agent of a chain can match a new arriving agent, the policy forms a new *chain-segment*, which is a maximal sequence of matches resulting from a *local search*, in which the next matched agent is selected randomly while breaking ties in favor of H agents (so the policy does not always identify the longest possible chain-segment, which requires a global search and may be computationally hard). We prove an upper-bound on the average waiting time that scales with $1/p_H$ for all positive arrival rates. We also find that even in the regime $\lambda_E > \lambda_H$ where the waiting time scales similarly under both matching technologies, chains result in lower waiting times than bilateral matching.

We provide comparative statics over the arrival rates of both types. We show (analytically for $p_E = 1$ and numerically for $p_E < 1$), that the average waiting time decreases when the arrival rate of either type increases. When $p_E = 1$, we further find that the average waiting time is independent of the constant d. Similar patterns hold numerically when $p_E < 1$. Finally, we are able to compute the average chain-segment (which plays an important operational role for example in kidney exchange). An increase in λ_E or d, decreases the average length of a chain-segment. In contrast, increasing λ_H has the opposite effect.

Next we provide brief intuition for some of the main findings, beginning with why the market composition and the desired matching technology are tightly connected. Under the bilateral setting when easy-to-match agents arrive more frequently, almost all hard-to-match agents will be matched with easy-to-match ones resulting in a scaling of $1/p_H$; on the other hand, when hard-to-match agents arrive more frequently, many of them will have to match with each other resulting in a scaling of $1/p_H^2$ which is the inverse of the probability that two H agents can match each other. In contrast, matching through chains does not require such "coincidence of wants" between pairs of H agents even when H agents are the majority. This results in a waiting time that scales with $1/p_H$ regardless of the composition. We further find that the heterogeneity in the marketplace may lead to non-trivial effects when increasing participation;

The intuition for why, in the bilateral setting, H agents may be harmed when attracting more H agents to the market is the following: When $\lambda_H < \lambda_E$, increasing λ_H reduces the chance that an existing H agent will match with the next arriving E agent. However, when $\lambda_H > \lambda_E$, increasing λ_H implies on one hand that more H agents must match with each other, and on the other hand, it reduces the time to form H-H matches. The first effect is initially stronger, but the second effect dominates once λ_H is sufficiently large.[6]

Understanding the impact of market composition by providing comparative statics requires us to not only compute the scaling of asymptotic behavior of average waiting time but also to characterize the exact limits. Such exact characterization in a heterogeneous model is particularly challenging as we need to analyze 2-dimensional Markov chains. For bilateral matching polices, we directly analyze the underlying 2-dimensional spatially non-homogeneous random walks. One of the main challenges in our analysis is the need to jointly bound the distribution in both dimensions, because applying methods such as Lyapunov functions or analyzing marginal probability distributions would not result in tight bounds. In doing so, we prove two auxiliary lemmas on concentration bounds for a general class of 2-dimensional random walks that can be of interest for studying similar random walks that may arise in other applications. For chain policies, we first couple the underlying Markov process with a 1-dimensional process where no E agent joins the market. Analysis of the resulting 1-dimensional Markov chain presents new challenges as transitions between non-neighboring states happen due to the possibility of forming arbitrarily long chain-segments. However, we show that the chain-segment formation process exhibits a memoryless property, which proves helpful in computing the waiting time limits.

[6] A similar effect happens in kidney exchange where O-A patient-donor pairs that cannot match with each other compete to match with scarce pairs with blood-type O donors. Note, however, that in our setting all agents can potentially match with each other; in particular this effect extends to sets of pairs that are blood type compatible with each other, like O-O pairs, some of which are much harder-to-match than others. We elaborate and provide intuition for other results throughout the paper.

5.1.1 Related work

A close stream of related papers study dynamic matching in models, in which agents' preferences are based on compatibility, i.e, agents are indifferent between whom they match with [4, 8, 126].

The impact of the matching technology is addressed in markets comprised of only easy-to-match agents [126] (with multiple coarse types) or only hard-to-match ones [8]. Ünver [126] finds that short cycles are sufficient for efficiency.[7] Anderson et al. [8] consider markets, in which all agents are ex ante symmetric and hard-to-match. They study the waiting-time scaling behavior of myopic policies that attempt to match each agent upon arrival in three settings of exchanges, 2-ways, 2 and 3-ways, and chains, and find that moving from 2-ways or 3-ways to chains significantly reduces the average waiting time.[8] Our paper bridges the gap by looking at a model with both hard- and easy-to-match agents and thus allowing for different levels of thickness in the market. [9]

The papers above also find that, by and large, myopic policies are near-optimal: Ünver [126] analyzes a kidney exchange model with different types and deterministic compatibility structure across types and finds that matching upon arrival is near optimal, even though some waiting with certain types to facilitate three-way exchanges adds some benefits. [10] Anderson et al. [8] consider a homogeneous model without departures (similar to our model with $\lambda_E = 0$) and finds that there is little benefit from waiting before matching under both matching technologies of short cycles and chains.[11] Akbarpour et al. [4] consider a homogeneous model with departures and finds that the optimality gap of the policy that matches without waiting remains constant as the match probability decreases. Moreover, using data-driven simulations, Ashlagi et al. [17] study the impact of match-run frequency,

[7] The findings by Ünver [126] thus provide a rationale for the static large market results (see, e.g. Roth et al. [107]).

[8] See also Dickerson et al. [43] that demonstrate the benefit of chains using simulations in dynamic kidney exchange pools.

[9] Ding et al. [49] study a similar two-type model in a static setting and quantifies the effectiveness of matching through chains taking a novel random walk approach.

[10] See also [62], who study a similar compatibility-based inventory control model.

[11] The waiting-time scales with the same factor with or without waiting before matching.

and show that among polices that match periodically (e.g., every week or every day), high matching frequencies perform best. [12] This paper builds on these findings, and only analyzes myopic policies that search for a match upon arrival of a new agent.

We elaborate on the relation to [8], which is closest to our paper. Studying myopic policies under a homogeneous setting resulted valuable insights. Some insights, however, do not carry over to heterogeneous settings like kidney exchange (See Subsection 1.2.1). For instance, merging markets is often sought as a solution to improve efficiency. A homogeneous model predicts that increasing arrival rates (or merging markets) will always decrease waiting times. In contrast, we find that merging heterogeneous markets may not decrease waiting times for both markets. The homogeneous model by Anderson et al. [8] predicts very infrequent but very long chain-segments. Our model predicts shorter chain-segments, which fits better empirical evidence (chain-segments typically consist of only a few pairs). Further, we remark that some questions cannot be addressed in a homogeneous setting; for instance kidney exchange programs attempt to attract easy-to-match pairs [12]; but the impact of such an increase cannot be investigated in a homogeneous model. As another example, exchange programs usually assign high priority to hard-to-match pairs; effect of such prioritization cannot be studied in a homogeneous model. Overall it is natural and important to study richer models in order to address relevant policy questions.

Another stream of related research considers models of agents' preferences that do not depend only on compatibility. These papers find that policies that match without waiting are inefficient [22, 50, 58, 70] since some waiting can improve the quality of matches.[13]

Our work is also related to the problem of matching multi-class customers to multi-class servers studied in queueing literature (e.g., Adan and Weiss [1], Caldentey et al. [34]).

In our model, an agent can be thought as a pair of customer-server, and the compatibility between any two agents is probabilistic, thus we will not have a finite number of queues.

Finally our work is related to the online matching literature that study online matching

[12] Non-myopic policies have also been studied, for example Dickerson et al. [44] study forward-looking polices by casting the dynamic matching problem as a high-dimensional dynamic program, and develop a heuristic to overcome the curse of dimensionality.

[13] See also related results in queueing models [30, 76].

in which the underlying graph is bipartite and agents on one side of the graph are all present in the market and only agents on the other side arrive over time [54, 61, 67, 72, 81].

5.1.2 Organization

In Section 5.2 we introduce the model, polices, and the underlying stochastic processes. In Subsection 1.2.1 we provide a brief background on kidney exchange further motivating our framework and study. In Section 5.3 we present the main theoretical results and Section 5.4 complements the results with numerical experiments. Section 5.5 outlines the main proof ideas and techniques along with the details of Markov chains induced by each policy. Section 5.6 concludes. For the sake of brevity, we only include proofs of selected results in the main text. The detailed proofs of the rest of the statements are deferred to clearly marked appendices.

5.2 Model

We study an infinite-horizon dynamic matching market, where each arriving agent is endowed with a single item she wants to exchange for another item she finds compatible. Agents are indifferent between compatible items and wish to exchange as early as possible, their cost of waiting being proportional to the waiting time.

There are two types of agents, H and E, referred by hard-to-match and easy-to-match, respectively. Beginning at time $t = 0$, agents of type $T \in \{H, E\}$ arrive to the market according to an independent Poisson process with rate $\lambda_T > 0$.

Any agent of type H (E) finds the item of any other agent compatible independently with probability p_H (p_E). Our analysis is asymptotic in $p_H \to 0$, while p_E is a fixed constant. So, on average, an H agent finds significantly fewer items compatible than an E agent. We say that an agent j is *matched* by agent i, if agent j receives agent i's item. An agent leaves the market only when she is *matched*, i.e., she receives a compatible item.

We study matching policies in two different settings, distinguished by how agents can exchange items. In the first setting two agents can exchange items bilaterally in a cyclic fashion. In the second setting agents exchange items through chains; at time $t = 0$, there are d special agents called *altruistic agents* who are willing to give an item without getting anything in return (all other agents that will arrive to the market are regular agents who want to exchange their item for another item).[14] Each agent in a chain receives a compatible item from one agent and gives to the next. At any given time, there are exactly $d \geq 1$ agents who are either altruistic or received an item but have not given their item. The latter are called *bridge agents*. We sometimes refer to altruistic agents also by bridge agents. The transactions between two bridge agents in a given chain is called a *chain-segment*. We assume that matches in a chain-segment are conducted instantaneously. A *policy* is a mapping from the history of exchanges and the state of the marketplace to a set of feasible exchanges involving non-overlapping sets of agents.

We adopt the average waiting time in steady-state as the measure of the efficiency of a policy (the waiting of an agent is the difference between her departure time and her arrival time). In our model, the average waiting time of one type of agents is equivalent to the average number of agents of that type in the marketplace divided by the arrival rate of that type since these two quantities are proportional to each other by Little's law.

It is convenient to think about the state of the marketplace at any time in terms of a *compatibility graph*, which is a directed graph with each agent represented by a node, and a directed edge from i to j means that agent j finds agent i's item compatible. Let $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$ denote the (observed) compatibility graph at time t. When a new agent arrives directed edges are formed in each direction independently and with probabilities corresponding to the agents' types, between the arriving agent and each agent in the marketplace. A bilateral exchange is a directed cycle of length two in the compatibility graph and a chain-segment is a directed path in this graph starting from a bridge or altruistic agent.

We study the following myopic policies, which attempt to match agents upon arrival.

Definition 3 (*BilateralMatch(T)* for $T \in \{H, E\}$)**.** Upon arrival of a new agent, if a cycle

[14] Having altruistic agents is an intrinsic property of the market in the sense that some markets do not have access to such agents.

of length 2 can be formed with the newly arrived agent, it is removed. If more than one such cycle exists, priority is assigned to cycles with agents of type T. Further ties are broken uniformly at random.

Definition 4 (*ChainMatch(d)*)**.** There are d bridge or altruistic agents in the market at any given time. We describe first the policy for $d = 1$. Consider a new arriving agent i_1. If i_1 does not have an incoming edge from the bridge agent then no matches happen. Otherwise, a chain-segment begins with matching i_1 by the bridge agent and advances as follows. First we search for an unmatched H agent that has an incoming edge from i_1; if there is one or more such H agents, we select one uniformly at random; otherwise, if no such an H exists, we search for an unmatched E agent that has an incoming edge from i_1 (again breaking ties uniformly at random). This process repeats itself immediately from the selected agent (selected agents cannot be reselected) until we reach an agent that cannot match any other agent, forming a disjoint path. All agents in the disjoint path leave the market except the last agent who then becomes a bridge agent.

When there are $d > 1$ altruistic/bridge agents, if there is at least one directed edge from one of them to the newly arrived agent, one of such edges is selected uniformly at random. As the process moves forward, each altruistic agent eventually gives her item to an arriving agent and starts a chain.

Under the *ChainMatch(d)* policy, upon arrival of a new agent a maximal chain-segment (path) is identified through local search originating from a bridge agent. [15] Note that the chain-segment has a positive length if and only if at least one bridge/altruistic agent has a directed edge to the new agent.

For brevity we often refer to *BilateralMatch(E)*, *BilateralMatch(H)*, and *ChainMatch(d)*, by $\mathcal{B}_H$, $\mathcal{B}_E$, and $\mathcal{C}(d)$, respectively. All the policies above are Markov policies, and thus define a continuous-time Markov chain (CTMC). The following observation will allow us to ignore the edges within the market when analyzing the underlying stochastic processes.

[15] Our local search chain-segment formation process bears similarity to Phase 1 of the two-phase clearing procedure of [49].

Observation 1. *For each policy $\mathcal{B}_H$, $\mathcal{B}_E$, and $\mathcal{C}(d)$, we can construct a 2-dimensional CTMC where states correspond to the number of waiting H and E agents, and which has a consistent evolution with that of the market under the corresponding policy.*

The observation is immediate for the bilateral policies $\mathcal{B}_H$ and $\mathcal{B}_E$; due to their myopic behavior there are no 2-length cycles in the market except with a new arriving agent, implying that the corresponding Markov chains can be fully specified using only the set of vertices. For the $\mathcal{C}(d)$ policy, the observation is more subtle. Note that under this policy there is no outgoing edge from a bridge agent to any waiting agent, again due to the myopic behavior of the policy. The first time we examine whether there is an edge from i to j, we effectively flip a bias coin with probability p_H (p_E) if the agent j is of type H (E). Importantly, we examine at most once whether a directed edge from i to j exists by the definition of the policy since i either leaves the market or becomes a bridge agent, in which case it will never match to j. Since both the edge formation and the matching policies do not depend on agents' identities (rather only on their types) we can merely keep track of the number of agents of each type.

In the remainder of the paper, for any policy $\mathcal{P}$, we focus on the simplified state space $\{[H_t^{\mathcal{P}}, E_t^{\mathcal{P}}]; t \geq 0\}$, which captures the number of hard- and easy-to-match agents at any time t, and we denote the corresponding transition rate matrix by $Q^{\mathcal{P}}$.

Given the *self-regulating* dynamic undergoing each matching process, one would expect that all three (irreducible) CTMC's reach steady-state. A rigorous statement and proof is provided in Appendix B.7. Hereafter, we are concerned only with steady-state analysis; For policy $\mathcal{P}$, we denote its steady-state distribution by $\pi^{\mathcal{P}}$. The random vector $[H^{\mathcal{P}}, E^{\mathcal{P}}]$ is the random number of H and E agents in steady-state, i.e., the vector is distributed according to distribution $\pi^{\mathcal{P}}$. Finally we define $w_H^{\mathcal{P}}$ ($w_E^{\mathcal{P}}$) to be the average waiting time of type H (E) agents under policy $\mathcal{P}$. Little's law implies that

$$w_H^{\mathcal{P}} = \frac{\mathbb{E}[H^{\mathcal{P}}]}{\lambda_H}, \text{ and } \quad w_E^{\mathcal{P}} = \frac{\mathbb{E}[E^{\mathcal{P}}]}{\lambda_E}. \tag{5.1}$$

Since in our model $p_H \to 0$ while p_E is kept constant, and all policies are myopic, one would expect that $w_E^{\mathcal{P}}$ is negligible compared to $w_H^{\mathcal{P}}$. We verify this claim below using numerical simulations and analytical proofs (see Figure 5-2 and Lemmas 13 and 14). Therefore we focus on analyzing the average waiting time of H agents under different policies.

In Section 5.3, we derive asymptotic results ($p_H \to 0$) for $w_H^{\mathcal{P}}$ for different set of parameters λ_H, λ_E and p_E. We note that $w_H^{\mathcal{P}}$ is indeed a function of four parameters, and a more precise notation would be $w_H^{\mathcal{P}}(\lambda_H, \lambda_E, p_H, p_E)$, but we drop these parameters for the sake of brevity.

5.3 Main results

We analyze the average waiting time under the myopic policies defined in Section 5.2. For bilateral matching polices, we identify a stark threshold in the scaling of waiting time when moving from the regime where a majority of arrivals are hard-to-match agents to the regime where the majority of arrivals are easy-to-match. Such a contrast does not exist when agents are matched through chains. We further study the impact of arrival rates of the two types on the market performance under the three polices.

5.3.1 Bilateral matching

This section considers the setting, in which agents match only through bilateral exchanges, i.e. through 2-way cycles.

Theorem 3. *Under the BilateralMatch(H) policy and in steady-state, the average waiting time $w_H^{\mathcal{B}_H}$ satisfies the following.*

- *If $\lambda_H < \lambda_E$, then* $\lim_{p_H \to 0} p_H w_H^{\mathcal{B}_H} = \frac{\ln\left(\frac{\lambda_E}{\lambda_E - \lambda_H}\right)}{p_E \lambda_H}$.
- *If $\lambda_H > \lambda_E$, then* $\lim_{p_H \to 0} p_H^2 w_H^{\mathcal{B}_H} = \frac{\ln\left(\frac{2\lambda_H}{\lambda_H + \lambda_E}\right)}{\lambda_H}$.

Theorem 3 provides not only the scaling laws on $w_H^{\mathcal{B}_H}$ but also the associated constants. The following corollaries provide comparative statics with respect to λ_H.

Corollary 1. *Consider the BilateralMatch(H) policy and fix λ_E. The limiting average waiting time $w_H^{\mathcal{B}_H}$ increases with λ_H in the interval $\lambda_H < \lambda_E$.*

Corollary 2. *Consider the BilateralMatch(H) policy and fix λ_E. The limiting average waiting time $w_H^{\mathcal{B}_H}$ increases with λ_H in the interval $\lambda_E < \lambda_H < x^*\lambda_E$, and decreases in the interval $\lambda_H > x^*\lambda_E$, where $x^* \approx 2.18$ is the unique solution of*

$$(x+1)\ln(2 - 2/(x+1)) = 1. \tag{5.2}$$

The above theorem and corollaries provide several messages on the impact of thickness on the performance of bilateral matching. First, the main factor in the asymptotic behavior of $w_H^{\mathcal{B}_H}$ is which type of agents has a larger arrival rate. Some intuition for the scaling factors is the following. Agents' average waiting time is inversely proportional to the probability of a bilateral match to occur. Under a myopic bilateral policy, no existing pair of agents in the market can match with each other. For an arriving H agent, the probability of forming a bilateral match with an existing E agent is $p_E p_H$, and with an existing H agent is p_H^2. When $\lambda_H < \lambda_E$, almost all H agents are matched with E agents resulting in an average waiting time that scales with $1/p_E p_H$. When H agents arrive more frequently than E agents, there are simply not enough E agents to match with H. So a non-negligible fraction of H agents match with each other and thus the scaling of the average waiting time increases to $1/p_H^2$.

Second, the arrival rates affect the average waiting times directly and not necessarily monotonically. Increasing the arrival rate of E agents always decreases the average waiting time. But this is not the case with H agents. When $\lambda_H < \lambda_E$, the average waiting time of H agents increases with λ_H. So in this regime where almost all H agents match bilaterally with E agents, increasing λ_H reduces the chance of an arbitrary H agent to match with the next E agent. When $\lambda_H > \lambda_E$, there is a non-monotone behavior of the waiting time when increasing λ_H. Increasing λ_H has two effects: (i) more H agents must match bilaterally with their same type, which is a negative effect, and (ii) for an existing H agent, it reduces the time to match with another H agent, which is a positive effect. After a certain threshold, the positive effect from having more H agents dominates the negative effect.

The key insight from the above discussion is that in a heterogeneous market, increasing the arrival rate does not always result in improving the waiting time since H agents impose a negative externality on other H agents under certain market compositions. This cannot be captured in a homogeneous model with only hard-to-match agents (the model studied in [8]).

Finally, we comment on the impact of p_E on the waiting time. When $\lambda_H < \lambda_E$, $w_H^{\mathcal{B}_H}$ is decreasing in p_E. On the other hand, when $\lambda_H > \lambda_E$, $w_H^{\mathcal{B}_H}$ is independent of p_E. The intuition is that in the former, all H agents match with E agents and in the latter the dominant factor in the average waiting time is due to 2-ways between H agents, which is independent of p_E.

The proof of Theorem 3 amounts to analyzing the underlying 2-dimensional continuous-time spatially non-homogeneous random walk. The description of the random walk is presented in Subsection 5.5.1 (Figure 5-8), along with a heuristic that helps us guess the right constants, and build intuition on the behavior of the random walk. The main idea behind the proof is establishing concentration results for a 2-dimensional CTMC where the steady-state distribution decays geometrically when moving away from the expectation. These concentration results allow us to establish matching lower and upper bounds on $w_H^{\mathcal{B}_H}$ (the proof is outlined in Subsection 5.5.1 with details deferred to Appendix B.2). We note that one of the main challenges in our analysis is the need to jointly bound the distribution in both dimensions, because analyzing marginal probability distributions would not result in tight bounds. As a byproduct of our analysis, in Subsection 5.5.2, we state two auxiliary lemmas on concentration bounds for a general class of 2-dimensional random walks. The corollaries follow from basic analysis of the corresponding constants (as a function of λ_H). Both corollaries are proved in Appendix B.2.2.

Theorem 4. *Under the BilateralMatch(E) policy and in steady-state, the average waiting time $w_H^{\mathcal{B}_E}$ satisfies the following.*

- *If $\lambda_H < \lambda_E$, then* $\frac{\ln\left(\frac{\lambda_E}{\lambda_E - \lambda_H}\right)}{p_E \lambda_H} \leq \lim_{p_H \to 0} p_H w_H^{\mathcal{B}_E} \leq \frac{\ln\left(\frac{2\lambda_E}{\lambda_E - \lambda_H}\right)}{p_E \lambda_H}$.
- *If $\lambda_H > \lambda_E$, then* $\lim_{p_H \to 0} p_H^2 w_H^{\mathcal{B}_E} = \frac{\ln\left(\frac{2\lambda_H}{\lambda_H + \lambda_E}\right)}{\lambda_H}$.

Comparing results of Theorems 3 and 4, we observe that when $\lambda_H < \lambda_E$, the average waiting time of H agents is larger or equal when prioritizing E agents rather then H agents (numerical simulations presented in Subsection 5.4.2 suggest that prioritizing E agents results in a strictly larger average waiting time). Nevertheless, the scaling remains the same. However, when $\lambda_H > \lambda_E$ prioritizing E agents does not impact the waiting time of H agents. The intuition is as follows. When $\lambda_H > \lambda_E$, the number of H agents waiting in the market scales as $1/p_H^2$, suggesting that the chance that an E agent does not match immediately upon arrival vanishes. Therefore assigning priority to E agents is redundant.[16]

The proof of Theorem 4 also requires analysis of the underlying 2-dimensional continuous-time spatially non-homogeneous random walk, and, in most parts, follows a similar structure to the proof of Theorem 3. A detailed description of the random walk is presented in Subsection 5.5.3. The proof of the upper and lower bounds is presented in Appendix B.3, where establishing the upper bound requires new ideas beyond the concentration results: we couple the Markov process underlying policy $\mathcal{B}_E$ with another process in which an E agent that cannot form a match upon arrival turns into an H agent.[17] In Subsection 5.5.3, we also provide a heuristic argument that leads us to guess that the exact limit is $\frac{\ln\left(\frac{\lambda_E+\lambda_H}{\lambda_E-\lambda_H}\right)}{p_E p_H}$ (See Figure 5-6 in Subsection 5.4.5).

5.3.2 Chain matching

In this section we analyze the *ChainMatch(d)* policy, under which agents match myopically through chains.

[16] We note that for neither policy *BilateralMatch(H)* nor *BilateralMatch(E)*, we are able to derive the exact waiting time behavior when $\lambda_E = \lambda_H$. However, our simulation results, presented in Appendix B.10, suggest that under both policies, the waiting time scales with $1/p_H$.

[17] In Subsection 5.5.3 we provide a rough intuition on why we cannot close the gap between our upper and lower bounds on $w_H^{\mathcal{B}_E}$ for the regime $\lambda_H < \lambda_E$.

Waiting time behavior

Theorem 5. *Let $d \geq 1$ be a constant (independent of p_H). Under the ChainMatch(d) policy and in steady-state, the average waiting time $w_H^{\mathcal{C}(d)}$ satisfies*

$$\lim_{p_H \to 0} p_H w_H^{\mathcal{C}(d)} \leq \frac{\ln\left(\frac{\lambda_H}{\lambda_E(1-(1-p_E)^d)} + 1\right)}{\lambda_H}.$$

The above theorem establishes an upper-bound on $w_H^{\mathcal{C}(d)}$ that scales with $1/p_H$. In Appendix B.9, we also establish a lower-bound on $w_H^{\mathcal{C}(d)}$ that scales with $1/p_H$. A stronger result is obtained for the special case, in which $p_E = 1$:

Proposition 2. *Let $p_E = 1$ and $d \geq 1$ be a constant (independent of p_H). Then*

$$\lim_{p_H \to 0} p_H w_H^{\mathcal{C}(d)} = \frac{\ln\left(\frac{\lambda_H}{\lambda_E} + 1\right)}{\lambda_H}.$$

Consequently, $\lim_{p_H \to 0} p_H w_H^{\mathcal{C}(d)}$ *decreases with λ_E and λ_H.*

First we discuss the intuition behind Proposition 2, which states that when $p_E = 1$, any constant number of altruistic agents will result in the same behavior of $w_H^{\mathcal{C}(d)}$. The positive impact of having $d > 1$ altruistic agents stems from the increase in probability of starting a new chain-segment. When an H agent arrives, the probability that she finds one of the bridge agents acceptable is $1 - (1 - p_H)^d$ which vanishes as $p_H \to 0$. When an E agent arrives she will always be matched by one of the bridge agents and proceed to advance the chain-segment, and thus there is no advantage in having more than one bridge agent.

For $p_E < 1$, we give a heuristic argument (in Appendix B.5), in which we analyze a related 3-dimensional random walk by artificially assuming that chain-segments advance according to an independent Poisson process with a very high rate μ (recall that under $\mathcal{C}(d)$ policy, chain-segments are formed and executed instantaneously upon arrivals). The heuristic provides an estimated waiting time that scales as $\ln\left(\frac{\lambda_H+\lambda_E}{\lambda_H(1-(1-p_H)^d)+\lambda_E}\right)/(p_H\lambda_H)$. In the limit when p_H approaches zero, the constant becomes $\ln\left(\frac{\lambda_H+\lambda_E}{\lambda_E}\right)/\lambda_H$ which is consistent with Proposition 2. Numerical simulations that are aligned with the result of the heuristic argument are presented in Subsection 5.4.5 (see Figure 5-7).

The heuristic argument in Appendix B.5 and simulation results of Figure 5-7 both suggest that $\lim_{p_H \to 0} p_H w_H^{\mathcal{C}(d)}(p_E) = \lim_{p_H \to 0} p_H w_H^{\mathcal{C}(d)}(p_E = 1)$.[18] This means that, in the limit where $p_H \to 0$, (1) the variability of $w_H^{\mathcal{C}(d)}(p_E)$ with respect to d or p_E is negligible compared to a term of order $\Theta(1/p_H)$, (2) for any $0 < p_E \leq 1$, $w_H^{\mathcal{C}(d)}(p_E)$ is decreasing in both λ_E and λ_H. Our simulation results verify these behaviors (see Section 5.4.3 and Figure 5-3; in particular, note that Figure 5-3 Bottom right illustrates that $w_H^{\mathcal{C}(d)}(p_E)$ does not vary with p_E).

The main intuition for why $w_H^{\mathcal{C}(d)}(p_E)$ does not vary with d or p_E can be summarized as follows. Under *ChainMatch(d)*, each E agent, immediately after being matched, initiates a sub-segment that includes a sequence of H agents, i.e., it has the form $E - H - \ldots - H$.[19] Denote the expected number of H agents in a sub-segment by $\mathbb{E}[\Sigma_E]$. Observe that with high probability an existing H agent is matched through such a sub-segment, because the probability that an arriving H agent starts a sub-segment of the form $H - H - \ldots - H$ is vanishing. Further, in steady state, the number of H agents who join the market (i.e., they do not match immediately), $\lambda_H(1 - p_H)^d$, must equal the number of agents who match through a sub-segment. Thus the departure rate of H agents from the market equals $\lambda_E \mathbb{E}[\Sigma_E] + o(1)$, where $o(1)$ corresponds to the event that an arriving H agent starts a sub-segment. As the result of this balance equation, we have $\mathbb{E}[\Sigma_E] = \lambda_H/\lambda_E + o(1)$, i.e., $\mathbb{E}[\Sigma_E]$ does not depend on p_E or d (up to a negligible additive factor of order $o(1)$).

Therefore, regardless of p_E and d, in expectation, each E agent “helps" to match the same number of H agents. The only difference between the cases $p_E = 1$ and $p_E < 1$ is the timing, in which E agents form sub-segments: when $p_E = 1$, an E agent forms a sub-segment immediately upon arrival. On the other hand, when $p_E < 1$, an arriving E agent will join the market with probability $(1 - p_E)^d$. In this case, such an E agent will start a sub-segment (of the form $E - H - \ldots - H$) after staying in the market for a random duration which we

[18] With a slight abuse of notation, we denoted the expected waiting time of H agents under $\mathcal{C}(d)$ policy with parameter p_E by $w_H^{\mathcal{C}(d)}(p_E)$.

[19] Note that when $p_E < 1$, a chain-segment can consist of multiple sub-segments that are initiated by some of the E agents waiting in the market.

denote by X. We can think of X as the "delay" in matching the H agents in the sub-segment that starts with the E agent. With high probability, X remains a constant as p_H approaches 0 (this follows from Lemma 16 in Appendix B.9, where we show that the expected waiting time of an E agent is a constant). Therefore, the delay caused by the E agent joining the market remains a constant. Consequently, the waiting time of H agents in a market with $p_E < 1$ is within $o(1/p_H)$ of its counterpart in a market with $p_E = 1$. In Appendix B.6, we provide further details for the above argument and build on this intuition to provide another heuristic argument to show that $\lim_{p_H \to 0} p_H w_H^{\mathcal{C}(d)}(p_E) = \lim_{p_H \to 0} p_H w_H^{\mathcal{C}(d)}(p_E = 1)$.

Finally we comment on the chain-segment formation process; *ChainMatch(d)* policy forms chain-segments employing a local search process and indeed our analysis relies on such chain-segment formation process. This begs the question of how much the waiting time improves if we employed a global search (that searches for the longest possible chain-segment). A precise comparison is beyond the scope of our work, however, we make the following remarks: (1) In Figure 5-4 of Subsection 5.4.3, we numerically study this questions, and we see that advancing chains locally results in a small loss in comparison to policies that search globally for the longest possible chain-segment. (2) The lower-bound on the waiting time of any anonymous Markovian policy (See [8] and Proposition 21 in Appendix B.8) implies that the scaling of H-agent waiting time cannot be smaller than $1/p_H$ (unless the policy makes E agents wait for a very long time, i.e., proportional to $1/p_H$); Theorem 5 shows that the local-search method already achieves such a scaling.

Under the *ChainMatch(d)* policy, the length of a chain-segment trigged by a newly arrived agent is unrestricted. As a result the underlying CTMC is significantly more complicated to analyze than those that arise from bilateral policies and we need other techniques to prove Theorem 5. In order to bound $w_H^{\mathcal{C}(d)}$, we couple the underlying Markov chain with a 1-dimensional chain, in which E agents that are not matched upon arrival leave the market immediately (Lemma 3). A key property used in the analysis of the coupled 1-dimensional chain is that chain-segment formation exhibits a memoryless property.[20] This is due to the local search process used to advance a chain-segment, which randomly selects the next agent

[20] This is different from the Markov property of the overall CTMC under $\mathcal{C}(d)$.

among all possible agents (favoring H agents). The proof is presented in 5.5.4. Finally, we note that for the special case $p_E = 1$, the original CTMC is a 1-dimensional chain for which we can prove matching upper and lower bounds on the limit of $w_H^{\mathcal{C}(d)}$.

Theorems 3 and 5 together highlight the importance of having altruistic agents that can initiate chains. In the regime $\lambda_H > \lambda_E$ comparing $w_H^{\mathcal{B}_H}$ and $w_H^{\mathcal{C}(d)}$ is straightforward as the former scales as $1/p_H^2$ but the latter only scales as $1/p_H$. The following corollary (proven in appendix B.4) states that in the regime $\lambda_H < \lambda_E$ where both $w_H^{\mathcal{B}_H}$ and $w_H^{\mathcal{C}(d)}$ scale as $1/p_H$, *ChainMatch(d)* performs better:

Corollary 3. *For any λ_H, λ_E, p_E, and d, if $\lambda_H < \lambda_E$, then:*

$$\lim_{p_H \to 0} p_H w_H^{\mathcal{C}(d)} < \lim_{p_H \to 0} p_H w_H^{\mathcal{B}_H}.$$

In Subsection 5.4.4 we further compare *BilateralMatch(H)* to *ChainMatch(d)* in order to understand the importance of attracting easy-to-match agents in markets that have limited access to altruistic agents.

Chain-segment length

We analyze here the expected length of chain-segments formed under the *ChainMatch(d)* policy.

While we focus on the average waiting time to measure efficiency, length of chain-segments also play a significant role on the operational efficiency of the market. In kidney exchange for example, executing a chain-segment takes time and bears the risk of match failures.[21] These practical considerations motivate extending the analysis to the limiting behavior of chain-segments.

First we define the chain-segment length. Let $[H_k^{\mathcal{C}(d)}, E_k^{\mathcal{C}(d)}]$ denote the (discrete-time) Markov chain embedded in the CTMC $[H_t^{\mathcal{C}(d)}, E_t^{\mathcal{C}(d)}]$ resulting from observing the system at

[21] In this stylized model, we abstract away from both of these effects.

arrival epochs.[22] Define:

$$L_k = H_k + E_k - H_{k+1} - E_{k+1} + 1,$$

and let L be its corresponding random variable in steady-state; if the arriving agent cannot be matched by the bridge agent, she will join the market, and therefore $L_k = 0$; otherwise, a chain-segment of length $L_k \geq 1$ will be formed. The following proposition characterizes the chain-segment length in the limit:

Proposition 3. *Under the ChainMatch(d) policy and in steady-state,*

$$\lim_{p_H \to 0} \mathbb{E}[L \mid L \geq 1] = \frac{\lambda_H + \lambda_E(1 - p_E)^d}{\lambda_E(1 - (1 - p_E)^d)} + 1.$$

The proof is presented in Appendix B.4. We note that the expected chain length is decreasing in both λ_E and d, but increasing in λ_H; intuitively with more E agents or more bridge agents, chain-segments will be formed at a higher rate and thus be shorter (for a fixed λ_H). However, increasing λ_H does not significantly impact the frequency of chain-segment formation, but given that more H agents join the market within two consecutive chain-segments, the length of the chain-segment grows.

5.4 Numerical studies

In this section, we present a set of numerical simulations that complement the theoretical results of the previous section. In Subsection 5.4.1 we look at how merging markets with different compositions affect each market. Subsection 5.4.2 explores the impact of giving priorities when using the bilateral matching policy. Subsection 5.4.3 presents comparative statics for chain matching when $p_E < 1$, and subsection 5.4.4 highlights the advantage of having chains. Finally, Subsection 5.4.5 compares our theoretical bounds (for cases for which we do not have matching upper and lower bounds) to heuristics guesses and simulations.

[22] Note that every time an agent arrives, the Markov chain advances in discrete time from k to $k + 1$.

All simulations in this section are conducted by first computing the average number of agents in the market; then applying Little's law (5.1). In order to compute the number of agents, we simulate the discrete-time Markov chain embedded in the corresponding CTMC resulting from observing the system at arrival epochs. We denote T the number of arrivals (not counting the d initial altruistic agents in the case of $\mathcal{C}(d)$). In order to remove the transient behavior, the numbers reported correspond to the time average over the second half of the simulation.

5.4.1 Merging markets

We consider here the effects from merging two markets, with arrival rates $(\lambda_{H,1}, \lambda_{E,1})$ and $(\lambda_{H,2}, \lambda_{E,2})$ under bilateral exchanges using the *BilateralMatch(H)* policy. This expands Theorem 3, which provides comparative statics in the limit when p_H tends to zero.

We consider two numerical examples to illustrate these effects. In both examples the arrival rates to the first market are kept fixed while the arrivals rates to the second market vary. For any pair of arrivals we compare the waiting time $w_{H,1}$ of H agents in the first market with the average waiting time $w_{H,1-2}$ in the merged market. The results are plotted in Figure 5-1. Consistent with our prediction, merging can result in one of the markets being worse off. Note that this can happen even if the majority type is the same for both markets (e.g., when $\lambda_{H,1} > \lambda_{E,1}$ and $\lambda_{H,2} > \lambda_{E,2}$). This highlights the effect of arrival rates beyond their impact on the scaling factor. [23]

5.4.2 Impact of priorities in bilateral matching

We compare here the average waiting time of H agents under the *BilateralMatch(H)* and *BilateralMatch(E)* policies. From Theorems 3 and 4 it follows that (i) when $\lambda_H > \lambda_E$, asymptotically, the average waiting time of H agents is the same under both policies, but (ii) when $\lambda_H < \lambda_E$, the average waiting time of H agents under $\mathcal{B}_H$ is at most the average

[23] We note that the constants computed in Theorem 3 allow us to determine whether market one is better off or worse off for any $(\lambda_{H,2}, \lambda_{E,2})$, and to compute the boundary separating the two regions, in the limit $p_H \to 0$.

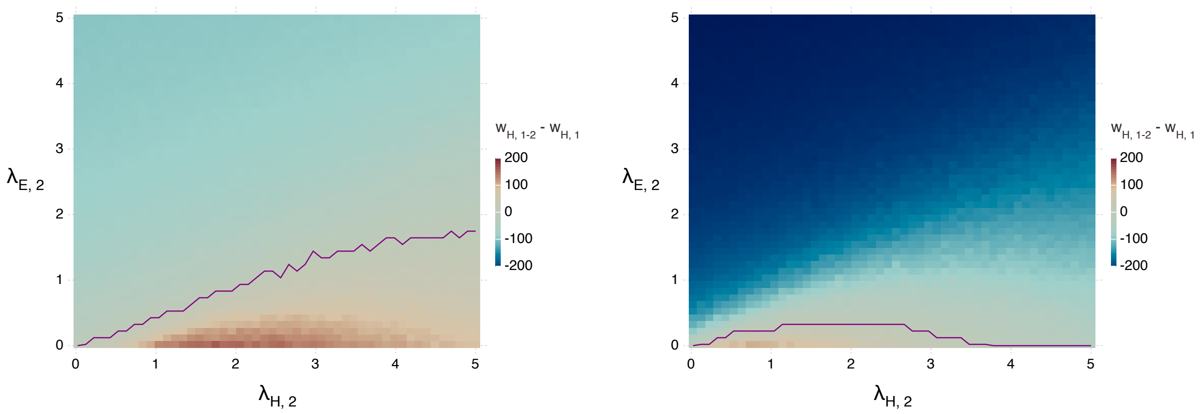

Figure 5-1: Change in the waiting time for H agents of the first market: $w_{H,1-2} - w_{H,1}$, as a function of $(\lambda_{H,2}, \lambda_{E,2})$, for $p_E = 0.5$, $p_H = 0.02$, after $T = 10^5$ iterations. Left subplot corresponds to $\lambda_{H,1} = 1, \lambda_{E,1} = 1.3$, right subplot corresponds to $\lambda_{H,1} = 1.3, \lambda_{E,1} = 1$. The purple line separates the region where the waiting time increases after merging (below the line) and the region where it decreases (above the line).

waiting time under $\mathcal{B}_E$. However, numerical simulations suggest that the average waiting time of H agents is indeed strictly smaller under $\mathcal{B}_H$ than under $\mathcal{B}_E$ (Figure 5-2 left). For instance, in simulation setting of Figure 5-2, when $\lambda_H = 4$ and $\lambda_E = 5$, we have $w_H^{\mathcal{B}_E} = 534$ while $w_H^{\mathcal{B}_H} = 388$. The average waiting times of E agents are plotted in Figure 5-2(right).

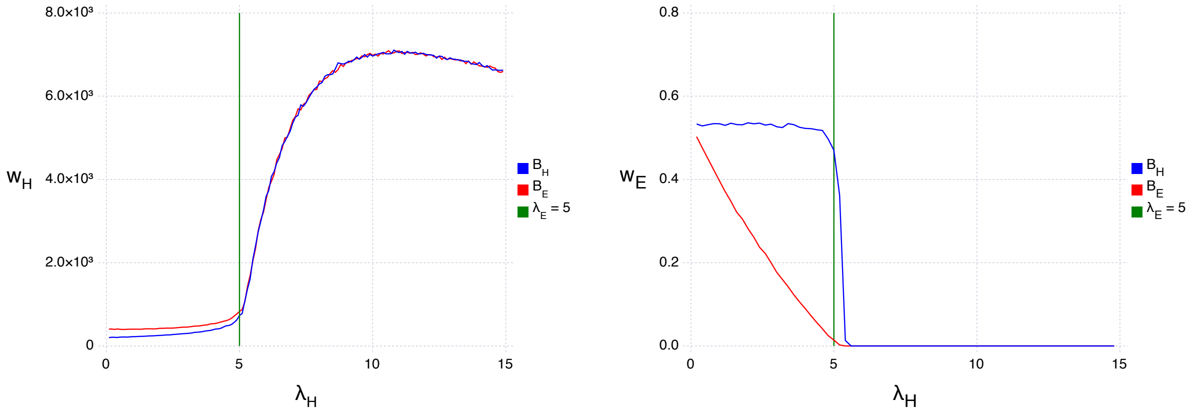

Figure 5-2: Comparison of w_H (left) and w_E (right) for $\mathcal{B}_H$, $\mathcal{B}_E$, as a function of λ_H, for a fixed $\lambda_E = 5$, $T = 2 \cdot 10^6$, $p_E = 0.5$ and $p_H = 0.002$.

The main insight is that the benefit from assigning priority to hard-to-match agents varies based on the composition of the market. Further, our qualitative insights can be useful in understanding the tradeoffs that may arise in markets where easy-to-match agents have outside options. For example, when $\lambda_H > \lambda_E$, there is no tradeoff from prioritizing E agents. This issue arises in kidney exchange, where very easy-to-match patient-donor pairs (such as compatible pairs) may choose to get transplanted elsewhere.

5.4.3 Comparative statics in chain matching with $p_E < 1$

We run simulations using *ChainMatch(d)* to numerically explore the effects that varying λ_E, λ_H, and d can have on $w_H^{\mathcal{C}(d)}$. We find that $w_H^{\mathcal{C}(d)}$ decreases as the arrival rate of either types increases (Figure 5-3 top left and top right). Moreover, the value of an additional altruistic agent also diminishes with increasing λ_E, λ_H or d. Further, as p_H decreases, the impact of d vanishes (Figure 5-3 bottom left). Recall from Proposition 2 that when $p_E = 1$, $w_H^{\mathcal{C}(d)}$ (i) decreases in λ_E, (ii) decreases in λ_H, and (iii) does not depend on d. Although proving (i)-(iii) for $p_E < 1$ remains an open problem, Figure 5-3 top left, top right, and bottom left show numerically (i)-(iii), respectively, when $p_E = 0.5$.

Further, the bottom right plot in figure 5-3 plots $w_H^{\mathcal{C}(d)}$ when p_E ranges from 0.1 to 1, and it shows that $w_H^{\mathcal{C}(d)}$ does not vary with p_E (we refer the reader to Appendix B.6, for a heuristic argument on why - in the limit - $w_H^{\mathcal{C}(d)}$ does not vary with p_E and d).

Next we study the loss from employing a local search for forming chain-segments rather than looking for the maximum-length path at each chain-segment-formation. For this, we define a new policy *Max-Chains* that upon starting a chain-segment searches for the chain-segment that maximizes lexicographically the number of H agents matched, while breaking ties over matching more agents over all.

We observe that the benefit of using *Max-Chains* is small when λ_H is small compared to λ_E, and it increases as λ_H increases. If we consider $\lambda_E/2 \leq \lambda_H \leq 2\lambda_E$ as the practical range relevant to the kidney exchange programs, our simulations suggest that the loss ranges between 5 to 15%.

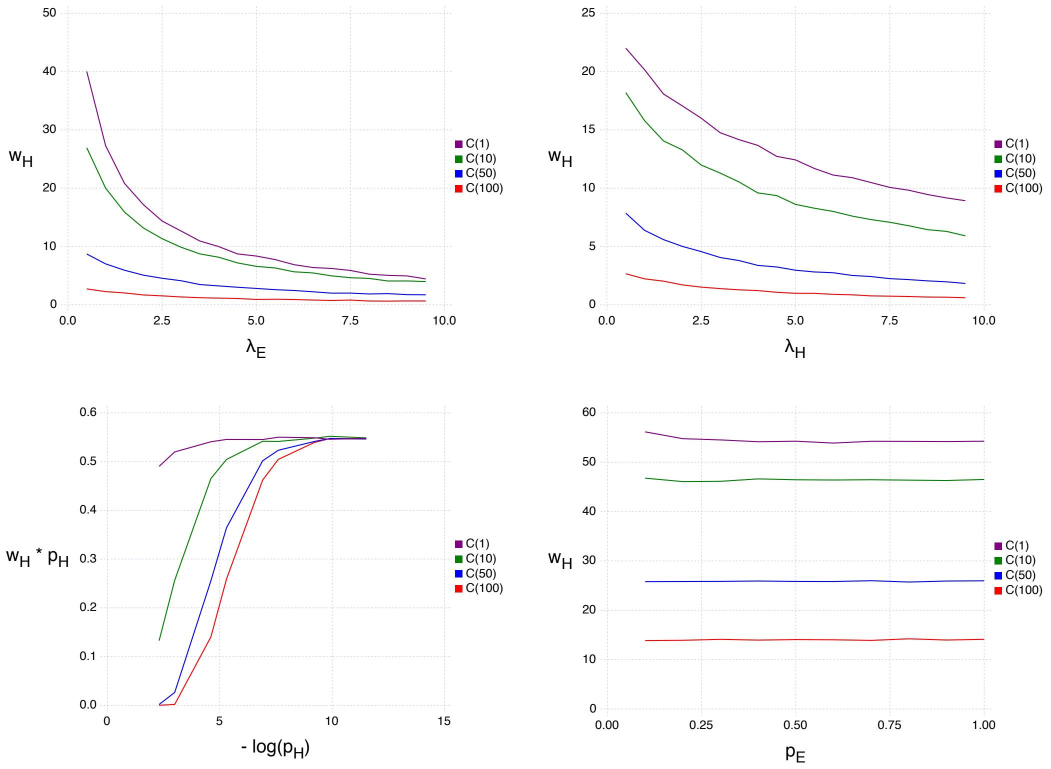

Figure 5-3: Top left: $w_H^{\mathcal{C}(d)}$ as a function of λ_E, for varying values of d, for fixed $\lambda_H = 2$, $T = 10^5$, $p_E = 0.5$, $p_H = 0.02$. Top right: $w_H^{\mathcal{C}(d)}$ as a function of λ_H, for varying values of d, for fixed $\lambda_E = 2$, $T = 10^5$, $p_E = 0.5$, $p_H = 0.02$. Bottom left: Normalized waiting times (i.e., $p_H w_H^{\mathcal{C}(d)}$) in the case of chains as a function of $-\log(p_H)$, for varying values of d, for fixed $\lambda_H = 2$, $\lambda_E = 1$, $T = 10^5$, $p_E = 0.5$. Bottom right: $w_H^{\mathcal{C}(d)}$ as a function of p_E, for different values of d, for fixed $\lambda_E = 1$, $\lambda_H = 2$, $T = 10^6$ and $p_H = 0.02$.

5.4.4 Impact of the matching technology: bilateral vs. chain matching

Theorems 3 and 5 imply that for *any* arrival rates (λ_H, λ_E) matching through chains even with only one initial altruistic agent (i.e., under *ChainMatch(1)*) results in shorter average waiting time for H agents. The theoretical gap is significant when $\lambda_H > \lambda_E$. We run numerical simulations for a variety of parameters to examine these differences (see Figure 5-5).

To further highlight the benefit of matching through chains, we consider the following

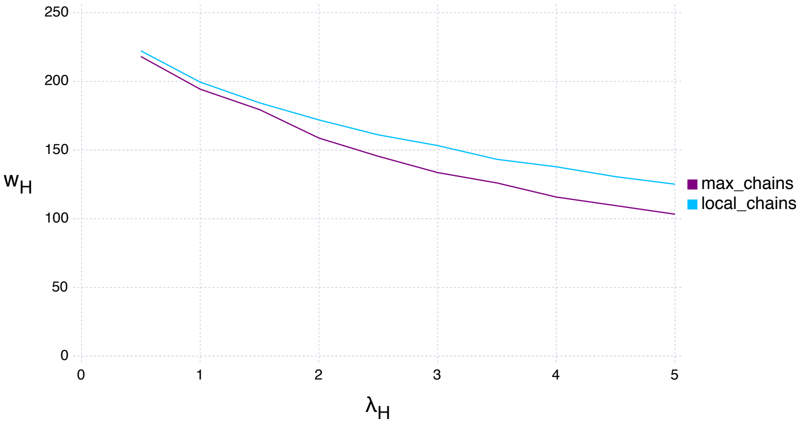

Figure 5-4: Waiting times w_H for chains conducted with local-search ($\mathcal{C}(d)$) and *Max-Chains* as a function of λ_H, for fixed $\lambda_E = 2$, $T = 10^5$, $p_E = 0.5$, $p_H = 0.002$.

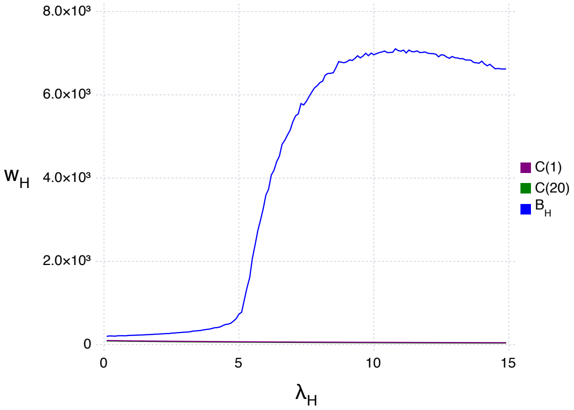

Figure 5-5: Comparison of w_H, for $\mathcal{B}_H$, $\mathcal{C}(1)$, and $\mathcal{C}(20)$, as a function of λ_H, for a fixed $\lambda_E = 5$, $T = 2 * 10^6$, $p_E = 0.5$ and $p_H = 0.02$.

scenario: Suppose market 1 has rates $(\lambda_{H,1}, \lambda_{E,1})$ with $\lambda_{H,1} < \lambda_{E,1}$ and is endowed with d altruistic agents and employs policy *ChainMatch(d)*. Now consider a second market with arrival rates $(\lambda_{H,2}, \lambda_{E,2})$ that does not have any altruistic agents and therefore employs *BilateralMatch(H)*. Further suppose $\lambda_{H,1} = \lambda_{H,2} = \lambda_H$; how many more E agents does market 2 need to attract to be able to compete with market 1 in term of average waiting times of H agents? In the limit $p_H \to 0$, by Theorems 3 and 5, for this to happens it is necessary that:

$$\frac{\ln\left(\frac{\lambda_H}{\lambda_{E,1}(1-(1-p_E)^d)}+1\right)}{\lambda_H} \geq \frac{\ln\left(\frac{\lambda_{E,2}}{\lambda_{E,2}-\lambda_H}\right)}{\lambda_H p_E},$$

which is equivalent to:

$$\lambda_{E,2} \geq \frac{\lambda_H(\lambda_H+\lambda_{E,1}(1-(1-p_E)^d))^{p_E}}{(\lambda_H+\lambda_{E,1}(1-(1-p_E)^d))^{p_E}-(\lambda_{E,1}(1-(1-p_E)^d))^{p_E}}.$$

Note that the above condition is only a necessary condition, and valid in the limit $p_H \to 0$. In the case where $p_E = 1$, Proposition 2 makes this also a sufficient condition, and it simplifies to $\lambda_{E,2} \geq \lambda_H + \lambda_{E,1}$. In table 5.1, we report the numerical values for $\lambda_{E,2}$ such that in simulations $w_{H,2}^{\mathcal{B}_H} = w_{H,1}^{\mathcal{C}(d)}$.

p_E	0.1	0.3	0.5	0.9	1.0
d = 1	20.75	8.45	5.4	3.3	3.0
d = 10	27.15	10.25	6.55	3.9	3.6
d = 50	66.05	24.8	15.1	9.0	8.15

Table 5.1: $\lambda_{E,2}$ as a function of p_E and d, for $p_H = 0.02$, $\lambda_H = 1$ and $\lambda_{E,1} = 2$, $T = 10^6$.

5.4.5 Theoretical bounds vs heuristics vs. simulation

In two cases, our theoretical results yield bounds which are are not tight. However, in each of these cases we generate a heuristic guess for the exact behaviour. We plot here the simulation results, our heuristically generated guess (described later in Section 5.5.3 and Appendix B.5) and the theoretical bounds for a variety of parameters. The first case is under the policy *BilateralMatch(E)* when $\lambda_H < \lambda_E$. Figure 5-6 shows that our heuristic analysis (described in Section 5.5.3) results in a guess of $\frac{\ln\left(\frac{\lambda_E+\lambda_H}{\lambda_E-\lambda_H}\right)}{p_E p_H}$ that coincides with the simulation results. The figure further illustrates the behavior of our theoretical bounds for different parameters.

The second case is under the policy *ChainMatch(d)* when $p_E < 1$. Here too, Figure 5-7 shows that that our heuristic guess $\ln\left(\frac{\lambda_H+\lambda_E}{\lambda_H(1-(1-p_H)^d)+\lambda_E}\right)/p_H$ (described in Appendix B.5) coincides with the numerical simulations.

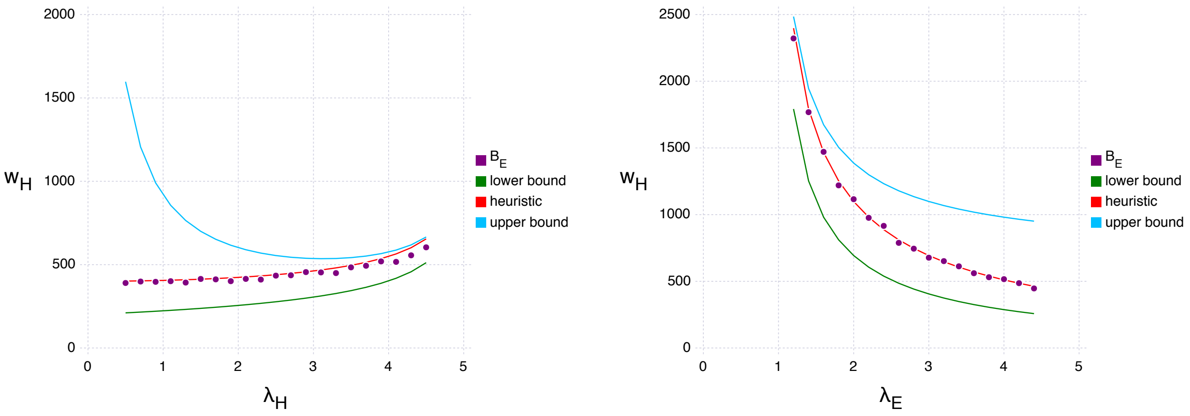

Figure 5-6: Left: $w_H^{\mathcal{B}_E}$ as a function of λ_H, for $\lambda_E = 5$, $T = 10^5$, $p_E = 0.5$, $p_H = 0.002$. Right: $w_H^{\mathcal{B}_E}$ as a function of λ_E, for $\lambda_H = 1$, $T = 10^5$, $p_E = 0.5$, $p_H = 0.002$.

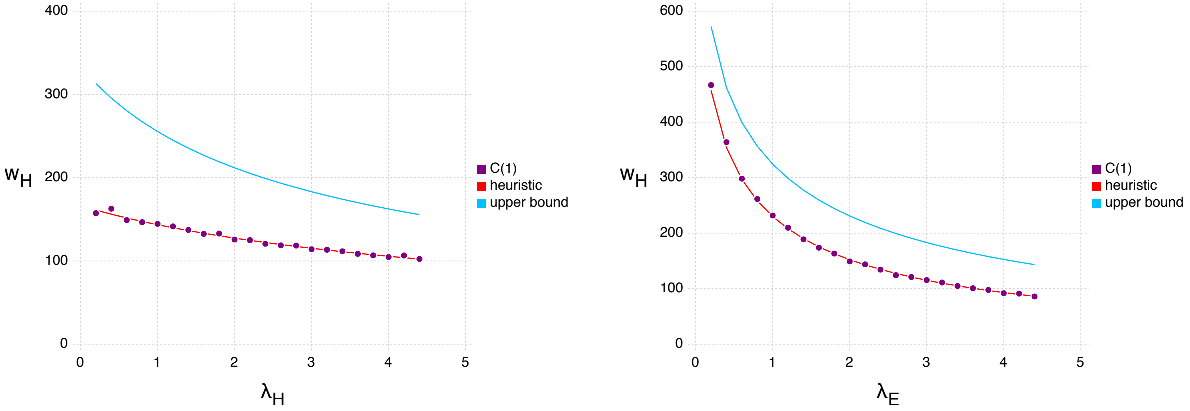

Figure 5-7: Left: $w_H^{\mathcal{C}(1)}$ as a function of λ_H, for $\lambda_E = 3$, $T = 10^5$, $p_E = 0.5$, $p_H = 0.002$. Right: $w_H^{\mathcal{C}(1)}$ as a function of λ_E, for $\lambda_H = 3$, $T = 10^5$, $p_E = 0.5$, $p_H = 0.002$.

5.5 Proof ideas and outline of analysis

The analysis of each policy follows a similar pattern, although technically analyzing the bilateral setting and the chain setting are very different. For bilateral policies, we first offer a heuristic that will help guessing the value of $\mathbb{E}[H^{\mathcal{P}}]$ (which is proportional to the average waiting time) and then proceed to rigourously analyze $\mathbb{E}[H^{\mathcal{P}}]$. For the chain policy, we first couple the underlying Markov chain with a 1-dimensional chain whose number of H agents serve as an upper-bound on the number of H agents under *ChainMatch(d)* policy. We then proceed to analyze the expected number of H agents in the coupled chain. In all three settings, the main idea is to prove that $H^{\mathcal{P}}$ is concentrated around $\mathbb{E}[H^{\mathcal{P}}]$ without directly computing the steady-state distribution, and based on the exponential decay of the tail distribution when moving away from the expected value.

We often use the following notations to avoid terms that vanish in the limit $p_H \to 0$. Let $f, g : [0,1] \to \mathbb{R}$; We write that $f = o(g)$ if $\lim_{p_H \to 0} \frac{f(p_H)}{g(p_H)} = 0$ and write that $f = O(g)$ if $\limsup_{p_H \to 0} \frac{f(p_H)}{g(p_H)} < \infty$.

5.5.1 The *BilateralMatch(H)* policy

In this section we analyze the policy $\mathcal{B}_H$, which forms myopically bilateral exchanges while prioritizing H agents. Under this policy, the evolution of the number of H and E agents in the market can be modeled by a CTMC $[H_t, E_t] \in \mathbb{N}^2$ with the following transition rates.

$$Q^{\mathcal{B}_H}([h,e],[h+1,e]) = \lambda_H(1-p_H^2)^h(1-p_Ep_H)^e \tag{5.3a}$$

$$Q^{\mathcal{B}_H}([h,e],[h-1,e]) = \lambda_H(1-(1-p_H^2)^h) + \lambda_E(1-(1-p_Ep_H)^h) \tag{5.3b}$$

$$Q^{\mathcal{B}_H}([h,e],[h,e+1]) = \lambda_E(1-p_Ep_H)^h(1-p_E^2)^e \tag{5.3c}$$

$$Q^{\mathcal{B}_H}([h,e],[h,e-1]) = \lambda_H(1-p_H^2)^h(1-(1-p_Ep_H)^e) + \lambda_E(1-p_Ep_H)^h(1-(1-p_E^2)^e)) \tag{5.3d}$$

The rates are computed based on the Poisson thinning property, simple counting arguments, and our assumption that edges are formed independently.

- Rightward rate (5.3a): moving from $[h, e]$ to $[h+1, e]$ happens when an H agent arrives and cannot form a cycle with any of the existing H agents (with probability $(1-p_H^2)^h$) nor with any of the existing E agents (with probability $(1 - p_E p_H)^e$).

- Leftward rate (5.3b): moving from $[h, e]$ to $[h - 1, e]$ happens when an H agent arrives and forms a cycle with at least one of the existing H agents (probability $(1-(1-p_H^2)^h)$) or an E agent arrives and forms a cycle with at least one of the existing H agents (probability $(1 - (1 - p_E p_H)^h)$).

- Upward rate (5.3c): moving from $[h, e]$ to $[h, e + 1]$ happens when an E agent arrives and cannot form a cycle with any of the existing H agents (probability $(1 - p_E p_H)^h$) nor with with any of the existing E agents (probability $(1 - p_E^2)^e$).

- Downward rate (5.3d): moving from $[h, e]$ to $[h, e-1]$ happens when an H agent arrives and cannot form a cycle with any of the existing H agent (probability $(1 - p_H^2)^h$) but can form a cycle with an existing E agents (probability $(1 - (1 - p_E p_H)^e)$), or an E agent arrives that cannot form a cycle with any of the existing H agents (probability $(1-p_E p_H)^h$) but can form a cycle with an existing E agent (probability $(1-(1-p_E^2)^e)$).

Note that the process is a 2-dimensional continuous-time spatially non-homogeneous random walk. Figure 5-8 illustrates this random walk along with its transition rates. Also observe that the leftward and downward rates (5.3b) and (5.3d) depend on the priority assigned to H agents, and these rates will change when prioritizing E agents, as we will see in the Subsection 5.5.3. However, fixing the priority, changing the tie-breaking rule between agents of the same type (for example, favoring agents with longer waiting times instead of selecting one at random) does not change the transition rates.

In Appendix B.7, we prove that the above (irreducible) CTMC is positive recurrent, and therefore reaches steady-state. This is intuitive given the above transition rates and the "self-regulating" behavior of the process. The larger the market, the larger the probability

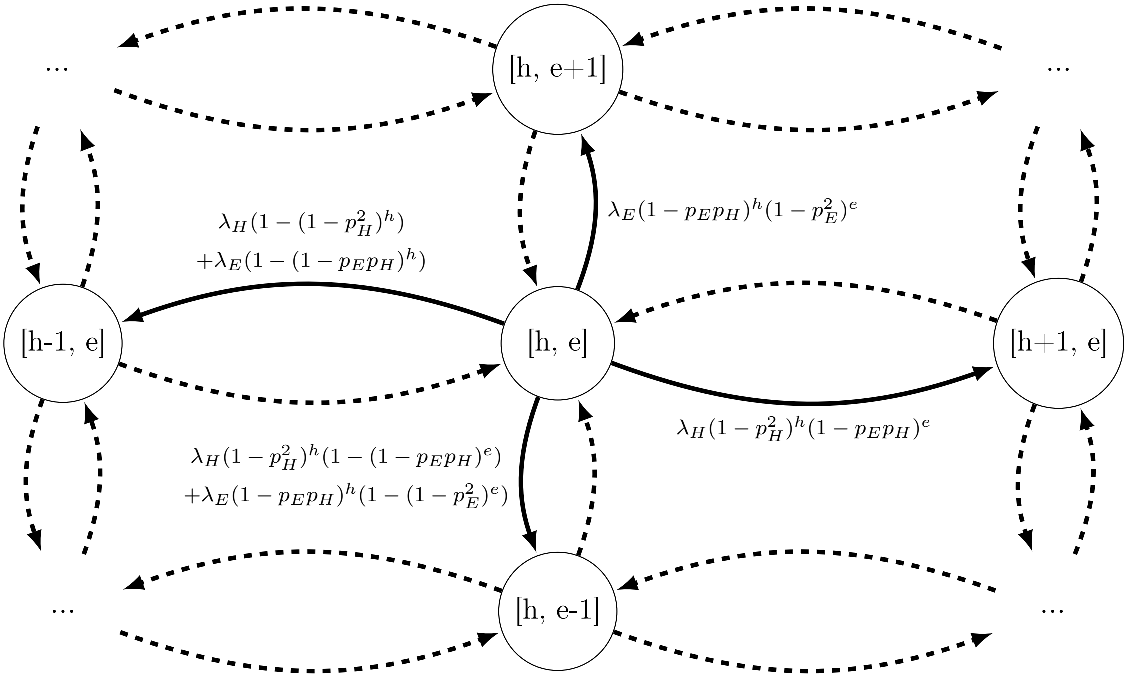
...
[h, e+1]
...
$\lambda_H(1-(1-p_H^2)^h)$
$+\lambda_E(1-(1-p_Ep_H)^h)$
$\lambda_E(1-p_Ep_H)^h(1-p_E^2)^e$
[h-1, e]
[h, e]
[h+1, e]
$\lambda_H(1-p_H^2)^h(1-p_Ep_H)^e$
$\lambda_H(1-p_H^2)^h(1-(1-p_Ep_H)^e)$
$+\lambda_E(1-p_Ep_H)^h(1-(1-p_E^2)^e)$
...
[h, e-1]
...

Figure 5-8: Transition rates (solid arrows) under the CTMC induced by the $\mathcal{B}_H$ policy.

that an arriving agent can form a cycle. Note that in steady-state, the expected drift for this CTMC in both horizontal and vertical dimension is zero.

The drifts are given in (5.3a)-(5.3d), and therefore,

$$\mathbb{E}\lambda_H(1-p_H^2)^{H^{\mathcal{B}_H}}(1-p_Ep_H)^{E^{\mathcal{B}_H}} - \lambda_H(1-(1-p_H^2)^{H^{\mathcal{B}_H}}) - \lambda_E(1-(1-p_Ep_H)^{H^{\mathcal{B}_H}}) = 0 \tag{5.4a}$$

$$\begin{aligned}&\mathbb{E}[\lambda_E(1-p_Ep_H)^{H^{\mathcal{B}_H}}(1-p_E^2)^{E^{\mathcal{B}_H}} - \lambda_H(1-p_H^2)^{H^{\mathcal{B}_H}}(1-(1-p_Ep_H)^{E^{\mathcal{B}_H}})\\ &\quad - \lambda_E(1-p_Ep_H)^{H^{\mathcal{B}_H}}(1-(1-p_E^2)^{E^{\mathcal{B}_H}})] = 0\end{aligned} \tag{5.4b}$$

Assuming that the random variables $H^{\mathcal{B}_H}$ and $E^{\mathcal{B}_H}$ are very concentrated around their expectations, a reasonable approximation is to move the expectation inside the functions and solve the above system of nonlinear equations, and thus obtain approximations for $\mathbb{E}[H^{\mathcal{B}_H}]$ and $\mathbb{E}[E^{\mathcal{B}_H}]$.

- For $\lambda_H < \lambda_E$, if we plug $[H^{\mathcal{B}_H}, E^{\mathcal{B}_H}] = \left[\frac{\ln\left(\frac{\lambda_E}{\lambda_E - \lambda_H}\right)}{p_E p_H}, \frac{-\ln(2)}{\ln(1-p_E^2)}\right]$ into (5.4a)-(5.4b), the right-hand sides will be $O(p_H)$ terms.
- For $\lambda_H > \lambda_E$, if we plug $[H^{\mathcal{B}_H}, E^{\mathcal{B}_H}] = \left[\frac{\ln\left(\frac{2\lambda_H}{\lambda_H + \lambda_E}\right)}{p_H^2}, 0\right]$ into (5.4a)-(5.4b), the right-hand sides will be $O(p_H^2)$ terms.

This heuristic exercise provides us the correct value of $\mathbb{E}[H^{\mathcal{B}_H}]$ in both cases. To establish this value rigorously and prove Theorem 3 we show, in the following two propositions, that $H^{\mathcal{B}_H}$ is highly concentrated around its mean.

Proposition 4. [Lower-bound] *Under $\mathcal{B}_H$ and in steady-state,*

- *If $\lambda_H < \lambda_E$, there exists a constant c_1 such that:*

$$\mathbb{P}\left[H^{\mathcal{B}_H} \leq \frac{1}{p_E p_H}\left(\ln\left(\frac{\lambda_E}{\lambda_E - \lambda_H}\right) - c_1 p_H^{1/4}\right)\right] \leq o(p_H).$$

- *If $\lambda_H > \lambda_E$, there exists a constant c_2 such that:*

$$\mathbb{P}\left[H^{\mathcal{B}_H} \leq \frac{1}{p_H^2}\left(\ln\left(\frac{2\lambda_H}{\lambda_E + \lambda_H}\right) - c_2\sqrt{p_H}\right)\right] \leq o(p_H).$$[24]

Proposition 5. [Upper-bound] *Under $\mathcal{B}_H$ and in steady-state, for any $k \geq 0$,*

- *If $\lambda_H < \lambda_E$, there exists a function $\gamma(p_H) = 1 - \sqrt{p_H} + o(\sqrt{p_H})$ and a constant c_3 such that:*

$$\mathbb{P}\left[H^{\mathcal{B}_H} \geq \frac{1}{p_E p_H}\left(\ln\left(\frac{\lambda_E}{\lambda_E - \lambda_H}\right) + c_3\sqrt{p_H}\right) + k\right] \leq \frac{\gamma(p_H)^k}{1 - \gamma(p_H)}.$$

- *If $\lambda_H > \lambda_E$, there exists a function $\gamma'(p_H) = 1 - \sqrt{p_H} + o(\sqrt{p_H})$ and a constant c_4 such that:*

$$\mathbb{P}\left[H^{\mathcal{B}_H} \geq \frac{1}{p_H^2}\left(\ln\left(\frac{2\lambda_H}{\lambda_E + \lambda_H}\right) + c_4\sqrt{p_H}\right) + k\right] \leq \frac{\gamma'(p_H)^k}{1 - \gamma'(p_H)}.$$

[24] While we do have closed form formulas for c_1, c_2 (similarly for c_3, c_4 defined in the next proposition), these values are not informative. We refer the reader to the proofs for more details.

Note that in both cases in Proposition 5, if $k = p_H^{-3/4}$ then the right-hand sides become $o(p_H^2)$.

The proof of Theorem 3 is a straightforward application of these propositions and the details are presented in Appendix B.2.1. To prove these propositions we derive exponentially decaying bounds on tails of the steady-state distribution of $H^{\mathcal{B}_H}$ and $E^{\mathcal{B}_H}$. In the next subsection we present two auxiliary lemmas that establish such bounds for a general class of 2-dimensional continuous-time random walks that includes the random walk defined above. The proof of Propositions 4 and 5 amount to applying these lemmas with appropriately defined parameters. The proofs are presented in Appendix B.2.4 and B.2.3, respectively.

5.5.2 Concentration bounds for a general class of 2-dimensional random walks

In the analysis of both *BilateralMatch(H)* and *BilateralMatch(E)* policies, we repeatedly bound the left-tail or the right-tail of the steady-state distribution of the number of H agents in the market. These bounds rely on certain properties of the corresponding 2-dimensional continuous-time random walks, which allow us to establish exponential decay on each tail of the steady-state distribution. To avoid repeating these concentration results for each particular setting, we take a unifying approach and state the following two auxiliary lemmas that establish concentration results for a general class of 2-dimensional random walks under certain conditions. These lemmas maybe useful in other applications that give rise to similar random walks.

Lemma 1. [Lower-bound] *Let $[X_t, Y_t] \in \mathbb{N}^2$ be a positive recurrent continuous time random walk with transition rate matrix Q and $[X, Y]$ be a corresponding random vector following its steady-state distribution. Suppose the following exist:*

Condition 1. A set $S \subset \mathbb{N}$ and a constant $\epsilon > 0$ such that $\mathbb{P}[Y \notin S] \leq \epsilon$.

Condition 2. A non-increasing function $f : \mathbb{N} \mapsto (0, \infty)$ such that $\forall y \in S$, $Q([x, y], [x + 1, y]) \geq f(x)$.

Condition 3. A non-decreasing function $g : \mathbb{N} \mapsto (0, \infty)$ such that $\forall y \in S$, $Q([x, y], [x - 1, y]) \leq g(x)$.

Then for all $\rho < 1$ and $\eta \in \mathbb{N}$ such that $\frac{g(\eta+1)}{f(\eta)} < \rho$, and any $k > 0$ we have:

$$\mathbb{P}[X \leq \eta - k] \leq \eta\epsilon \left(1 + \frac{1}{f(\eta) - g(\eta + 1)}\right) + \frac{\rho^k}{1 - \rho}.$$

Proof of Lemma 1. Let $\pi(x, y)$ be the joint distribution of $[X, Y]$, and consider he marginal distribution of X: $\pi_X(x) = \sum_{y \geq 0} \pi(x, y)$. In steady-state, conservation of flow implies:

$$\sum_{y \in S} \pi(x + 1, y)Q([x + 1, y], [x, y]) + \sum_{y \notin S} \pi(x + 1, y)Q([x + 1, y], [x, y])$$
$$= \sum_{y \in S} \pi(x, y)Q([x, y], [x + 1, y]) + \sum_{y \notin S} \pi(x, y)Q([x, y], [x + 1, y]).$$

Using Conditions 2 and 3, we upper-bound the left hand side and lower-bound the right hand side which results in having:

$$g(x + 1)\mathbb{P}[X = x + 1, Y \in S] + \mathbb{P}[X = x + 1, Y \notin S] \geq f(x)\mathbb{P}[X = x, Y \in S].$$

Let $\pi_S(x) = \mathbb{P}[X = x, Y \in S] = \sum_{y \in S} \pi(x, y)$. Observe that by Condition 1 we have: $\pi_X(x) \leq \pi_S(x) + \epsilon$. Using the fact that g is non-decreasing and f is non-increasing, we get for $x \leq \eta$:

$$\pi_S(x) \leq \frac{g(x + 1)}{f(x)}\pi_S(x + 1) + \frac{\mathbb{P}[Y \notin S]}{f(x)} \leq \rho\pi_S(x + 1) + \frac{\epsilon}{f(\eta)}.$$

We can subtract $\frac{\epsilon/f(\eta)}{1-\rho}$ from both sides and iterate: for all $j \geq 0$,

$$\pi_S(\eta - j) - \frac{\epsilon/f(\eta)}{1 - \rho} \leq \rho^j \left(\pi_S(\eta) - \frac{\epsilon/f(\eta)}{1 - \rho}\right) \leq \rho^j.$$

This allows us to conclude that for any $k > 0$:

$$\mathbb{P}[X \leq \eta - k] = \sum_{i=0}^{\eta-k} \pi_X(i) \leq (\eta - k)\epsilon + \sum_{j=k}^{\eta} \pi_S(\eta - j)$$
$$\leq (\eta - k)\epsilon \left(1 + \frac{1/f(\eta)}{1-\rho}\right) + \sum_{j=k}^{\eta} \rho^j \leq \eta\epsilon \left(1 + \frac{1}{f(\eta)(1-\rho)}\right) + \frac{\rho^k}{1-\rho}.$$

□

Lemma 2. [Upper-bound] *Let $[X_t, Y_t] \in \mathbb{N}^2$ be a positive recurrent continuous time random walk with transition rate matrix Q and let $[X, Y]$ be a corresponding random vector following its steady-state distribution. Suppose the following exist:*

Condition 1. A mapping $S : \mathbb{N} \mapsto 2^{\mathbb{N}}$ and two constants $c \in \mathbb{R}^+, \delta \in (0, 1)$ such that $\mathbb{P}[Y \notin S(x)] \leq c\delta^x$.

Condition 2. Two functions $f, g : \mathbb{N} \mapsto (0, \infty)$ such that $\forall y \in S$, $Q([x, y], [x + 1, y]) \leq f(x)$ and $Q([x, y], [x - 1, y]) \geq g(x)$.

Then for all $\eta > 0$ and $\rho \in [\delta, 1)$ such that $\forall x \geq \eta$, $\frac{f(x)}{g(x+1)} \leq \rho$, and $\frac{\delta^x}{g(x+1)} \leq \frac{\rho^x}{g(\eta+1)}$, and for any $k > 0$ we have: [25]

$$\mathbb{P}[X \geq \eta + k] \leq \frac{\rho^k}{1-\rho} \left(1 + c + \frac{c(k + 1)}{g(\eta + 1) - f(\eta)}\right).$$

The proof of Lemma 2 follows similar arguments to that of Lemma 1 and is deferred to Appendix B.1.

[25] Note that the above conditions are weaker than that of Lemma 1 (where f is non-increasing, g is non-decreasing and $\frac{f(\eta)}{g(\eta+1)} \leq \rho$). We will need this for the proofs of Propositions 4 and 19 where the corresponding function g is not monotone.

5.5.3 The *BilateralMatch(E)* policy

The policy $\mathcal{B}_E$ forms myopically bilateral exchanges while prioritizing E agents. The transition rate of the underlying CTMC are as follows.

$$Q^{\mathcal{B}_E}([h,e],[h+1,e]) = \lambda_H(1-p_H^2)^h(1-p_Ep_H)^e \tag{5.5a}$$
$$Q^{\mathcal{B}_E}([h,e],[h-1,e]) = \lambda_H(1-p_Ep_H)^e(1-(1-p_H^2)^h) + \lambda_E(1-p_E^2)^e(1-(1-p_Ep_H)^h) \tag{5.5b}$$
$$Q^{\mathcal{B}_E}([h,e],[h,e+1]) = \lambda_E(1-p_Ep_H)^h(1-p_E^2)^e \tag{5.5c}$$
$$Q^{\mathcal{B}_E}([h,e],[h,e-1]) = \lambda_H(1-(1-p_Ep_H)^e) + \lambda_E(1-(1-p_E^2)^e) \tag{5.5d}$$

The rates are computed similarly to those under the *BilateralMatch(H)*. Observe that prioritizing E results in different leftward and downward rates (5.5b) and (5.5d) than the corresponding rates under *BilateralMatch(H)*. In particular note that in the leftward rate (moving from $[h,e]$ to $[h-1,e]$), the probability that an arriving E agent matches an existing H agent depends now on the current number of E agents. This dependency does not exist in *BilateralMatch(H)*. This makes the analysis of *BilateralMatch(E)* more difficult since we need to compute tight bounds also on the number of E agents in the market. While we are able to prove such bounds in the case $\lambda_H > \lambda_E$, we are not able to do so in the case $\lambda_H < \lambda_E$.

As before, we set the expected drifts at steady-state in both dimensions to zero, resulting in the following system of equations.

$$\begin{aligned}&\mathbb{E}[\lambda_H(1-p_H^2)^{H^{\mathcal{B}_E}}(1-p_Ep_H)^{E^{\mathcal{B}_E}} - \lambda_H(1-p_Ep_H)^{E^{\mathcal{B}_E}}(1-(1-p_H^2)^{H^{\mathcal{B}_E}})\\ &- \lambda_E(1-p_E^2)^{E^{\mathcal{B}_E}}(1-(1-p_Ep_H)^{H^{\mathcal{B}_E}})] = 0\end{aligned} \tag{5.6a}$$
$$\mathbb{E}\left[\lambda_E(1-p_Ep_H)^{H^{\mathcal{B}_E}}(1-p_E^2)^{E^{\mathcal{B}_E}} - \lambda_H(1-(1-p_Ep_H)^{E^{\mathcal{B}_E}}) - \lambda_E(1-(1-p_E^2)^{E^{\mathcal{B}_E}})\right] = 0. \tag{5.6b}$$

Similar to the heuristic analysis for *BilateralMatch(H)*, we can obtain the following approximations for $\mathbb{E}[H^{\mathcal{B}_E}]$ and $\mathbb{E}[E^{\mathcal{B}_E}]$.

- For the case $\lambda_H < \lambda_E$, if we plug $\left[\frac{\ln\left(\frac{\lambda_E+\lambda_H}{\lambda_E-\lambda_H}\right)}{p_E p_H}, \frac{\ln\left(\frac{\lambda_E+\lambda_H}{2\lambda_E}\right)}{\ln(1-p_E^2)}\right]$ into (5.6a)-(5.6b), the right-hand sides will be $O(p_H)$ terms.
- For the case, $\lambda_H > \lambda_E$, if we plug $\left[\frac{\ln\left(\frac{2\lambda_H}{\lambda_H+\lambda_E}\right)}{p_H^2}, 0\right]$ into (5.6a)-(5.6b), the right-hand sides will be $O(p_H^2)$ terms.

As stated in Theorem 4, for the case $\lambda_H > \lambda_E$ the constant for the limit of $\frac{\mathbb{E}[H^{\mathcal{B}_E}]}{\lambda_H}$ coincides with the solution given by the above heuristic. For the case, $\lambda_H < \lambda_E$, the constant resulting from the above heuristic argument lies in between the constants of the lower and upper bounds we can prove (in Theorem 4), i.e.,

$$\frac{\ln\left(\frac{\lambda_E}{\lambda_E-\lambda_H}\right)}{p_E} \leq \frac{\ln\left(\frac{\lambda_E+\lambda_H}{\lambda_E-\lambda_H}\right)}{p_E} \leq \frac{\ln\left(\frac{2\lambda_E}{\lambda_E-\lambda_H}\right)}{p_E}.$$

In Figure 5-6 (Section 5.4.5), we numerically show that $\frac{\ln\left(\frac{\lambda_E+\lambda_H}{\lambda_E-\lambda_H}\right)}{p_E}$ is indeed the right constant.

The proof of the case $\lambda_H > \lambda_E$, and the lower bound when $\lambda_H < \lambda_E$ in Theorem 4 follows similar steps as that of Theorem 3, and it uses the concentration results of the lemmas stated in the previous subsection. The difficulty in closing the gap between our lower and upper bounds for the case $\lambda_H < \lambda_E$ comes from the dependency of the leftward rate on the current number of E agents (i.e., the second term in (5.5b)). Our bounds on the right-tail of the distribution of number of E agent are not tight enough to result in a matching lower and upper bounds. Closing this gap remains an open question. A notable difference is that in (5.3a) and (5.3b), knowing that E is bounded above by a constant (independent of p_H) is enough to get matching upper and lower bounds (up to a vanishing term). This, however, is not the case in (5.5b). To prove the upper bound in the case $\lambda_H < \lambda_E$ we couple the Markov process underlying policy $\mathcal{B}_E$ with another process in which an E agent that cannot form a match upon arrival turns into an H agent. See subsection B.3.2.

5.5.4 The *ChainMatch(d)* policy

This section proves Theorem 5 and Proposition 2. As we could establish only an upper bound for the average waiting time when $p_E < 1$, we refer the reader to Appendix B.5 for a heuristic analysis that leads us to guess the constant that we can numerically verify to be the correct one (See Figure 5-7 of Subsection 5.4.5).

Instead of directly analyzing the *ChainMatch(d)* policy under our setting, we consider a modified setting, in which an E agent that does not match immediately upon arrival is removed from the system. We refer to this new setting under the policy *ChainMatch(d)* by $\hat{\mathcal{C}}(d)$. Observe that $H_t^{\hat{\mathcal{C}}(d)}$ is a 1-dimensional CTMC with the following transition rates:

$$Q^{\hat{\mathcal{C}}(d)}(h, h+1) = \lambda_H (1 - p_H)^d, \tag{5.7a}$$

$$Q^{\hat{\mathcal{C}}(d)}(h, h-i) = \Lambda (1 - p_H)^{h-i} \prod_{j=0}^{i-1} (1 - (1 - p_H)^{h-j}), \qquad \text{for } i \in \{1, 2, \ldots, h\}. \tag{5.7b}$$

Where we introduced the following notation: $\Lambda = \lambda_H(1 - (1 - p_H)^d) + \lambda_E(1 - (1 - p_E)^d)$, which is the rate at which a new chain-segment (possibly of length 1) starts, regardless of the current state, and let S_h be the random number of agents removed from the system, starting from state h.[26] The first expression, (5.7a), corresponds to rate, at which an H agent arrives, but cannot be matched by a bridge agent. The second expression, (5.7b), corresponds to the rate, at which an agent arrives, is matched by a bridge agent and forms a chain-segment of length i.[27] In Appendix B.7, we show that the above CTMC reaches steady-state.

For any $i \leq h$ we can write

[26] Note that using the notation from Section 5.3.2, $S_h + 1$ corresponds to the length of the chain-segment L_k for the 1-dimensional Markov chain.

[27] Observe that the case $i = 0$ is possible, and corresponds to an arriving agent that can receive from a bridge agent but cannot continue the chain further. In that case the CTMC does not transition and we consider the chain-segment to have length 1.

$$Q^{\hat{\mathcal{C}}(d)}(h, h-i) = \Lambda \mathbb{P}[S_h = i] = \Lambda(1-p_H)^{h-i} \prod_{j=0}^{i-1} (1-(1-p_H)^{h-j}). \tag{5.8}$$

Observe that we have

$$\mathbb{P}[S_h \geq k] = \prod_{j=0}^{k-1} (1-(1-p_H)^{h-j}) \tag{5.9}$$

The proof proceeds by showing that $\mathbb{E}[H^{\hat{\mathcal{C}}(d)}]$ serves as an upper bound for $\mathbb{E}[H^{\mathcal{C}(d)}]$ (Lemma 3) and then computing the limit of $\mathbb{E}[H^{\hat{\mathcal{C}}(d)}]$ (Proposition 6). Before that, we make the following crucial observation: the process of chain-segment formation under $\hat{\mathcal{C}}(d)$ exhibits a memoryless property. That is, for any state h and any $\tilde{i} \leq i \leq h$:

$$\begin{aligned} \mathbb{P}[S_h = i] &= (1-p_H)^{h-i} \prod_{j=0}^{i-1} (1-(1-p_H)^{h-j}) \qquad (5.10) \\ &= \prod_{j=0}^{\tilde{i}-1} \left(1-(1-p_H)^{h-j}\right) \left[(1-p_H)^{(h-\tilde{i})-(i-\tilde{i})} \prod_{j=0}^{(i-\tilde{i})-1} \left(1-(1-p_H)^{(h-\tilde{i})-j}\right)\right] \\ &= \mathbb{P}[S_h \geq \tilde{i}]\mathbb{P}[S_{h-\tilde{i}} = i-\tilde{i}], \end{aligned}$$

In other words, the event of forming a chain-segment of length i can be decomposed into two independent events: forming a chain-segment of length at least $\tilde{i}$ and then forming a chain-segment of length $i-\tilde{i}$ starting with $h-\tilde{i}$ agents in the market. This heavily relies on the fact that chain-segments proceed in a local search (one by one) fashion and the independence assumption. Indeed, the chain-segment formation in the original 2-dimensional chain $\mathcal{C}(d)$ has a similar property.

We now show that $\mathbb{E}[H^{\hat{\mathcal{C}}(d)}]$ is an upper bound for $\mathbb{E}[H^{\mathcal{C}(d)}]$.

Lemma 3. *The expected number of H agents in steady-state under $\hat{\mathcal{C}}(d)$ satisfies:*

$$\mathbb{E}[H^{\mathcal{C}(d)}] \leq \mathbb{E}[H^{\hat{\mathcal{C}}(d)}].$$

Proof. The proof is based on a coupling argument. Consider two copies of the arrival process, one under the setting of $\mathcal{C}(d)$ and one under $\hat{\mathcal{C}}(d)$. Let $[H_k^{\mathcal{C}(d)}, E_k^{\mathcal{C}(d)}]$ and $H_k^{\hat{\mathcal{C}}(d)}$ denote the embedded discrete-time Markov chain resulting from observing the two dynamic systems at arrival epochs. We prove a stronger result: at any step k, $H_k^{\mathcal{C}(d)} \leq H_k^{\hat{\mathcal{C}}(d)}$. We prove this using the following coupling:

1. Upon arrival of an H agent we flip a biased coin with probability $(1 - p_H)^d$. If the coin flip is head, the H agent cannot start a chain-segment, and both $H_{k+1}^{\mathcal{C}(d)}$ and $H_{k+1}^{\hat{\mathcal{C}}(d)}$ increment by one. If the coin flip is tail, the H agent starts a chain-segment in both systems. Suppose that $[H_k^{\mathcal{C}(d)}, E_k^{\mathcal{C}(d)}] = [h, e]$ and $H_k^{\hat{\mathcal{C}}(d)} = \tilde{h}$, and let $[L_{[h,e]}^H, L_{[h,e]}^E]$ denote the random number of H and E agents in the chain-segment formed under $\mathcal{C}(d)$ at state $[h, e]$; similarly let $S_{\tilde{h}}$ be the length of chain-segment formed under $\hat{\mathcal{C}}(d)$ at state $(\tilde{h})$. We dinstinguish between three cases:

   (a) $\tilde{h} \geq h$ and the event $\{S_{\tilde{h}} < (\tilde{h} - h)\}$ occurs: we let $[L_{[h,e]}^H, L_{[h,e]}^E]$ be realized independently of $S_{\tilde{h}}$.

   (b) $\tilde{h} \geq h$ and the event $\{S_{\tilde{h}} \geq (\tilde{h} - h)\}$ occurs. In this case the memoryless property of $\hat{\mathcal{C}}(d)$ in (5.10) can be rewritten as: $\mathbb{P}[S_{\tilde{h}} = i \mid S_{\tilde{h}} \geq (\tilde{h} - h)] = \mathbb{P}[S_h = i - (\tilde{h} - h)]$. This divides the chain-segment formation into two independent events: a subchain-segment of length $(\tilde{h} - h)$ is formed, and then a subchain-segment of length ξ, where ξ is a random variable drawn from the distribution of S_h.

      Now we focus on the chain-segment formation under $\mathcal{C}(d)$. Because H agents get a higher priority, the chain-segment can be computed in steps. Starting with $[h, e]$ agents, we first look for a subchain-segment L_1^H consisting of only H agents. When this chain-segment cannot be continued further with only H agents, we look for an E agent to continue the chain. If this happens (with probability $(1 - (1 - p_E)^e)$, we look for a second subchain-segment L_2^H of only H agents, etc.

Note that the first subchain-segment L_1^H also has the same distribution as S_h. We can therefore set $L_1^H = \xi$. All further subchain-segments L_i^H are realised independently.

(c) $\tilde{h} < h$: we let $[L_{[h,e]}^H, L_{[h,e]}^E]$ and $S_{\tilde{h}}$ be realized independently.[28]

2. Upon arrival of an E agent we flip a biased coin with probability $(1-p_E)^d$. If the flip is head, the E agent cannot start a chain-segment in either system and we have: $H_{k+1}^{\mathcal{C}(d)} = H_k^{\mathcal{C}(d)}$, and $H_{k+1}^{\hat{\mathcal{C}}(d)} = H_k^{\hat{\mathcal{C}}(d)}$; on the other hand, if the flip is tail, the E agent starts a chain-segment in both systems. The chain-segment formation in this case is exactly the same as the one for an H arrival.

Having the above coupling, we finish the proof by induction: The base case $k=0$ is trivial: $H_0^{\mathcal{C}(d)} = H_0^{\hat{\mathcal{C}}(d)} = 0$. Suppose $H_k^{\mathcal{C}(d)} \leq H_k^{\hat{\mathcal{C}}(d)}$ holds for k, we show that it also holds for $k+1$: if an H/E arrival does not start a chain-segment then by coupling construction $H_{k+1}^{\mathcal{C}(d)} \leq H_{k+1}^{\hat{\mathcal{C}}(d)}$. If an H arrival does start a chain-segment then we are either in Case (1a) or (1b). In the former the length of the chain-segment in $\hat{\mathcal{C}}(d)$ was not even long enough to bring the number of H agents back to $H_k^{\mathcal{C}(d)} = h$; therefore $H_{k+1}^{\mathcal{C}(d)} \leq H_{k+1}^{\hat{\mathcal{C}}(d)}$ holds. In the latter case, again by coupling construction $L_{[h,e]}^H \geq S_h + (\tilde{h} - h)$, which implies that $H_{k+1}^{\mathcal{C}(d)} \leq H_{k+1}^{\hat{\mathcal{C}}(d)}$ holds. A similar argument holds if an E arrival starts a chain-segment.

□

The next proposition computes $\mathbb{E}[H^{\hat{\mathcal{C}}(d)}]$ in the limit. Together with Lemma 3, this completes the proof of Theorem 5.

Proposition 6. *Under $\hat{\mathcal{C}}(d)$ and in steady-state, the expected number of H agents satisfies:*

$$\lim_{p_H \to 0} p_H \mathbb{E}[H^{\hat{\mathcal{C}}(d)}] = \ln\left(1 + \frac{\lambda_H}{\lambda_E(1-(1-p_E)^d)}\right).$$

[28] This case is only defined here for the sake of completeness, the induction will ensure that this never happens.

Proof. Let π be the steady-state probability distribution. By the conservation of flow from state h to $h+1$, we have:

$$\pi(h)\lambda_H(1-p_H)^d = \sum_{k\geq 1}\pi(h+k)\left(\sum_{i\leq h} Q^{\hat{\mathcal{C}}(d)}(h+k,i)\right).$$

Note that $\sum_{i\leq h} Q^{\hat{\mathcal{C}}(d)}(h+k,i)$ is the total leftward flow starting from state $h+k$ and ending at state $i \leq h$. Using (5.8) and (5.9), we have:

$$\sum_{i\leq h} Q^{\hat{\mathcal{C}}(d)}(h+k,i) = \Lambda\mathbb{P}[S_{h+k}\geq k],$$

and therefore,

$$\pi(h)\lambda_H(1-p_H)^d = \Lambda\sum_{k\geq 1}\pi(h+k)\mathbb{P}[S_{h+k}\geq k]. \tag{5.11}$$

Observe that applying Definition (5.9), we have $\mathbb{P}[S_{h+k}\geq k] = \mathbb{P}[S_{h+k}\geq k-1]\mathbb{P}[S_{h+1}\geq 1]$. Therefore we can rewrite (5.11) as:

$$\pi(h)\lambda_H(1-p_H)^d = \Lambda\left(\pi(h+1)\mathbb{P}[S_{h+1}\geq 1] + \mathbb{P}[S_{h+1}\geq 1]\sum_{k\geq 2}\pi(h+k)\mathbb{P}[S_{h+k}\geq k-1]\right). \tag{5.12}$$

Similarly we write the conservation of flow from state $h+1$ to $h+2$:

$$\begin{aligned}\pi(h+1)\lambda_H(1-p_H)^d &= \Lambda\sum_{k\geq 1}\pi(h+1+k)\mathbb{P}[S_{h+k+1}\geq k] \qquad (5.13)\\ &= \Lambda\sum_{k'\geq 2}\pi(h+k')\mathbb{P}[S_{h+k'}\geq k'-1],\end{aligned}$$

where the last step follows from a change of variable $k' = k+1$. Note that the summation in the RHS of (5.13) also appears in the second term of RHS of (5.12). Substituting $\sum_{k'\geq 2}\pi(h+k')\mathbb{P}[S_{h+k'}\geq k'-1]$ with $\pi(h+1)\lambda_H(1-p_H)^d/\Lambda$ in (5.12) gives that

$$\pi(h)\lambda_H(1-p_H)^d = \pi(h+1)\mathbb{P}[S_{h+1} \geq 1]\left(\Lambda + \lambda_H(1-p_H)^d\right). \tag{5.14}$$

We can now compute $\mathbb{E}[H^{\hat{\mathcal{C}}(d)}]$ by proving an upper and lower bound separately. We use the fact that for states far enough from the expectation, the distribution decays geometrically. We start with the upper-bound. Let $\eta = \ln\left(1 + \frac{\lambda_H}{\lambda_E(1-(1-p_E)^d)}\right)/p_H + 1/\sqrt{p_H}$. We know from (5.9) that $\mathbb{P}[S_{h+1} \geq 1] = 1-(1-p_H)^{h+1}$. This implies that for $h \geq \eta$,

$$\begin{aligned}\mathbb{P}[S_{h+1} \geq 1] &\geq 1-(1-p_H)^{\eta+1}\\ &= 1-e^{(\eta+1)\ln(1-p_H)}\\ &= 1-\frac{\lambda_E(1-(1-p_E)^d)}{\lambda_H+\lambda_E(1-(1-p_E)^d)}(1-\sqrt{p_H})+o(p_H),\end{aligned}$$

where we used the Taylor expansion $\ln(1-x) = -x - x^2/2 - x^3/3 - \ldots$.

Using (5.14) for $h \geq \eta$, we have:

$$\begin{aligned}\frac{\pi(h+1)}{\pi(h)} &= \frac{\lambda_H(1-p_H)^d}{\mathbb{P}[S_{h+1} \geq 1]\left(\Lambda+\lambda_H(1-p_H)^d\right)}\\ &\leq \frac{\lambda_H(1-p_H)^d}{\lambda_H+\lambda_E(1-(1-p_E)^d)\sqrt{p_H}}+o(\sqrt{p_H})\\ &= 1-c\sqrt{p_H}+o(\sqrt{p_H}) =: \delta,\end{aligned} \tag{5.15}$$

where $c = \frac{\lambda_E(1-(1-p_E)^d)}{\lambda_H}$. Having (5.15), we upper bound $\mathbb{E}[H^{\hat{\mathcal{C}}(d)}]$ as follows:

$$\begin{aligned}\mathbb{E}[H^{\hat{\mathcal{C}}(d)}] &= \sum_{h \le \eta + p_H^{-3/4}} \pi(h) + \sum_{h \ge \eta + p_H^{-3/4}+1} \pi(h) \\ &\le \eta + p_H^{-3/4} + \pi(\eta) \frac{\delta^{p_H^{-3/4}+1}}{(1-\delta)} \\ &= \frac{\ln\left(1 + \frac{\lambda_H}{\lambda_E(1-(1-p_E)^d)}\right)}{p_H} + o(1/p_H).\end{aligned}$$

Similarly we lower bound $\mathbb{E}[H^{\hat{\mathcal{C}}(d)}]$: let $\hat{\eta} = \ln\left(1 + \frac{\lambda_H}{\lambda_E(1-(1-p_E)^d)}\right)/p_H - 1/\sqrt{p_H}$, we can find $\hat{c}$ such that for $h \le \hat{\eta}$:

$$\frac{\pi(h)}{\pi(h+1)} \le 1 - \hat{c}\sqrt{p_H} + o(\sqrt{p_H})$$

The above inequality combined with Markov inequality enables us to lower bound $\mathbb{E}[H^{\hat{\mathcal{C}}(d)}]$ as follows:

$$\mathbb{E}[H^{\hat{\mathcal{C}}(d)}] \ge (\eta - p_H^{-3/4})\left(1 - \sum_{h=0}^{\eta - p_H^{-3/4}} \pi(h)\right) = \frac{\ln\left(1 + \frac{\lambda_H}{\lambda_E(1-(1-p_E)^d)}\right)}{p_H} + o(1/p_H).$$

□

Finally, note in the special case $p_E = 1$, an arriving E agent is matched immediately by a bridge agent, implying that $E_t^{\mathcal{C}(d)} = 0$ and $H_t^{\mathcal{C}(d)} = H_t^{\hat{\mathcal{C}}(d)}$; consequently Proposition 6 implies the limit stated in Proposition 2.

5.6 Final comments

In matching markets where monetary transfers are not allowed, exogenous thickness increases exchange opportunities [102]. Using a simple dynamic model with heterogeneous agents we find a tight connection between market thickness and the desired matching technology; matching through chains is significantly more efficient than (simple) bilateral matching only when the market is sufficiently thin. Furthermore, increasing the arrival rate of hard-to-match agents may have, under bilateral matching, an adverse effect on such agents.

An important dynamic matching market is kidney exchange, which enables incompatible patient-donor pairs to exchange donors. While our stylized model abstracts away from many details in this market, our findings may provide some useful insights to policy issues. When merging markets, which is an ongoing effort in various countries (see Section 1.2.1), or attracting different types of pairs, there may be negative effects on some pairs. This effect is well known for pairs with O patients and non-O donors who compete to match with scarce O donors in the pool [107]; our findings suggest that this negative effect extends also to blood-type compatible pairs (like O-O), many of which have very highly sensitized patients. Understanding these externalities are a key step towards aligning incentives towards cooperation between the relevant players [13]. Our findings further provide some insights about tradeoffs from prioritizing different types of pairs.

Next we discuss some limitations and possible extensions. One interesting challenge is to quantify the exact loss from restricting attention to myopic policies that do not wait before matching, rather than finding the optimal Markovian policy that may make some agents wait in order to increase matching opportunities.

[29] Another interesting direction is to extend the model to allow departures.[30] Finally, our focus has been on marketplaces, in which any pair of agents have a non-zero probability of forming a match. We found that the composition of the market crucially impacts the efficiency of the market. An interesting direction for future research would be to extend this study to two-sided marketplaces, in particular explore what features determine waiting times; for example, whether it is more beneficial to be on the short side or have a large ex ante match probability.

[29] Similar to [8] we can show our policies achieve the same scaling as the best anonymous Markovian policy (see Proposition 21 in Appendix B.8) but characterising the best constants is an open question.

[30] For example, [4] allow agents to depart prior to being matched and consider the match rate as the measure for efficiency.

Part II

Beyond Myopic Policies: Algorithms for Stochastic and Worst Case Graphs

Chapter 6

Maximizing Efficiency in Dynamic Matching Markets

6.1 Introduction

We study the problem of matching agents who arrive to a marketplace over time and leave after a short period. Agents can only be matched while they are present in the marketplace. There is a different value for matching every pair of agents, which does not vary over time. The planner's goal is to maximize the total value over a given finite time horizon.

Several marketplaces face a such a problem. Ride-hailing platforms have to match passengers with drivers, in which case the value of a match may depend on to the distance between the driver and the passenger. Such platforms may also carpool passengers and hence match passengers with each other, in which case the value of a match can represent the reduction in total distance traveled by the two matched passengers, compared to the distance traveled in individual rides. Kidney exchange platforms face the problem of matching incompatible patient-donor pairs with each other. In this context the value of a match can represent, for example, the quality adjusted life years due to the transplant. The common challenge in all these applications comes from the uncertainty associated with future arrivals and potential future matches.

We study matching algorithms that perform well across any sequence of arrivals, when there is no a priori information about the match values or arrival times. The underlying graph structure may be arbitrary and is not necessarily bipartite. Agents can be matched at any moment between their arrival and their departure. In that sense, our framework differs from the classic online matching literature where matching decisions have to be made immediately upon the arrival of an agent.

One important assumption we make is that each agent departs from the market exactly d time periods after her arrival. In the case of the carpooling application one may think of d as a (self-imposed) service requirement, which ensures that no passenger waits for too long before being matched. In that case, after d periods the passenger is assigned to an individual ride. Later on, we relax this assumption to allow for stochastic departures. It is worth noting that when departure times are allowed to be arbitrary, the competitive ratio of any algorithm is unbounded.

Contributions

Next we describe our basic graph-theoretic model and contributions. Time is discrete and one vertex of a given graph arrives at each time period. Every edge has a non-negative weight, representing the reward from matching these two vertices. A vertex cannot match more than d periods after its arrival; after d units of time the vertex becomes critical and departs. It is helpful to think of d as a service quality set by the platform and a passenger is assigned to a single ride after waiting for d periods of time.

The goal is to find a weighted matching with a large total weight in an online manner. This means that the decision for every vertex has to be made no later than d periods after its arrival (this differs from the classic online bipartite matching literature, in which $d = 0$). There is no a priori information about weights or arrival times and the underlying graph may be arbitrary and hence *non-bipartite*.

Our first results are given in a setting, in which the vertices arrive in an **adversarial order**. We introduce for this setting a $^1\!/_4$-competitive algorithm, termed *Postponed Greedy* (PG). We further show that no algorithm achieves a competitive ratio that is higher than $^1\!/_2$.

The key idea behind PG is to look at a virtual bipartite graph, in which each vertex is duplicated into a "buyer" and a "seller" copy. We enforce that the seller copy does not match before the vertex becomes critical. This enables us to postpone the matching decision until we have more information about the graph structure and the likely matchings. We then proceed in a manner similar to [55]: tentatively match each new buyer copy to the seller that maximizes its margin, i.e., the difference between edge weight, and the value of the seller's current match.

We extend the model to the case where the departure of vertices are determined stochastically. We show that when the departure distribution is memoryless and realized departure times are revealed to the algorithm just as becoming critical, one can adapt the PG algorithm to achieve a competitive ratio of $^1\!/_8$. It is worth noting that when departure times are chosen in an adversarial manner no algorithm can achieve a constant competitive ratio.

Related literature

There is a growing literature related to ride-sharing. Santi et al. [113] finds that about 80% of rides in Manhattan could be shared by two passengers. Many studies focus on rebalancing or dispatching problems without pooling, e.g., Banerjee et al. [24], Pavone et al. [97], Santi et al. [113], Spieser et al. [121], Zhang and Pavone [134]. Alonso-Mora et al. [6] studies real-time high-capacity ride-sharing. It does not consider, however, a graph-theoretic online formulation for matching rides.

This paper is closely related to the online matching literature. In the classic problem, introduced in Karp et al. [72], the graph is bipartite with vertices on one side waiting, while others are arriving sequentially and have to be matched *immediately* upon arrival. This work has numerous extensions, for example to stochastic arrivals and in the adwords context [54, 61, 67, 81, 83]. See [82] for a detailed survey. Our contributes to this literature in three ways. First, we provide algorithms that perform well on edge-weighted graphs. Second, our graph can be non-bipartite, which is the case in ride-sharing and kidney exchange. Third, all vertices can arrive over time and may remain for some given time until they are matched. Closely related is Huang et al. [66], which studies a similar model to ours in the non-weighted case, but allow departure times to be adversarial.

Several papers consider the problem of dynamic matching in the edge-weighted case. Feldman et al. [55] find that in the classic online bipartite setting, no algorithm achieves a constant approximation. They introduce a *free disposal* assumption, which allows to discard a matched vertex in favor of a new arriving vertex. They find, based on an algorithm by Lehmann et al. [75], that a greedy algorithm that matches a vertex to the highest marginal vertex, is 0.5-competitive. We build on this result for a special classes of bipartite graphs. In the adversarial setting Ashlagi et al. [16], Emek et al. [53] study the problem of minimizing the sum of distances between matched vertices and the sum of their waiting times. In their model no vertex leaves unmatched and our model does not account for vertices' waiting times. Few papers consider the stochastic environment [22, 65, 95]. These papers find that some waiting before matching is beneficial for improving efficiency.

Related to our work are some papers on job or packet scheduling. Jobs arrive over online to a buffer, and reveal upon arrival the deadline by which they need to be scheduled. The algorithm can schedule at most one job per time and the value of scheduling a job is independent from the time slot. Constant approximation algorithms are given by Chin et al. [36] and Li et al. [77].

Finally, there is a growing literature that focuses on dynamic matching motivated from kidney exchange [9, 18, 46, 126]. These papers focus mostly on random graphs with no weights. Closer to our paper is [5], which finds that in a sparse random graph, knowledge about the departure time of a vertex is beneficial and matching a vertex only when it becomes critical performs well. Our work differs from these papers in two ways: we consider the edge-weighted case, and, we make no assumption on the graph structure.

6.2 Model

Consider a weighted graph G with n vertices indexed by $i = 1, \dots n$. Vertices arrive sequentially over n periods and let $\sigma(i)$ denote the arrival time of vertex i. Let $v_{ij} \geq 0$ denote the weight on the undirected edge (i, j) between vertices i and j.

For vertices i and j with $\sigma(i) < \sigma(j)$, the weight v_{ij} on the edge between i and j is observed only after vertex j has arrived.

For $d \geq 1$, the **online graph with deadline** d, denoted by $G_{d,\sigma}$, has the same vertices as G, and the edge between i and j in G exists if an only if $|\sigma(i) - \sigma(j)| \leq d$. We say that i becomes **critical** at period $\sigma(i) + d$, at which time the online algorithm needs to either match it and collect the associated edge weight, or let it *depart* from the graph.

In this Chapter, we study the Adversarial Order (AO) setting, i.e. we assume that the order of arrivals is chosen by the adversary: $\sigma(i) = i$. In Chapter 7, we will study a similar model in the Random Order (RO) setting, where we assume that σ is sampled uniformly at random among all possible permutations S_n of $[1, n]$.

The goal is to find an online algorithm that generates a matching with high total weight. More precisely, we seek to design a randomized online algorithm that obtains in expectation a high fraction of the expected maximum-weight of a matching over $G_{d,\sigma}$.

To illustrate a natural tradeoff, consider the example in Figure 6-1 for $d = 1$. At period 2 the planner can either match vertices 1 and 2 or let vertex 1 remain unmatched. This simple example shows that no deterministic algorithm can obtain a constant competitive ratio. Furthermore, no algorithm can achieve a competitive ratio higher than $1/2$.

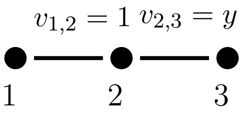

Figure 6-1: Let $d = 1$. Therefore, there is no edge between vertices 1 and 3. The algorithm needs to decide whether to match 1 with 2 and collect $v_{1,2}$ without knowing y.

The example in Figure 6-1 illustrates a necessary condition for the algorithm to achieve a constant competitive ratio: with some probability, vertex 2 needs to forgo the match with vertex 1. We ensure this property by assigning every vertex to be either a *seller* or a *buyer*. We then prevent sellers from matching before they become critical, while we allow buyers to be matched at any time.

It will be useful to first study a special case, in which the underlying graph G is bipartite, with sellers on one side and buyers and in the online graph a buyer and a seller cannot match if the buyer arrives before the seller. For such online graphs we show that a greedy algorithm given by Feldman et al. [55] is 0.5-competitive. We then build on this algorithm to design a randomized $1/4$-competitive algorithm for arbitrary graphs.

6.3 Main Results

6.3.1 An algorithm for bipartite constrained online graphs

Let G be a bipartite graph and σ be the order of arrivals. The online graph $G_{d,\sigma}$ is called **constrained bipartite** if for every seller s and every buyer b, there is no edge between s and b if $\sigma(b) < \sigma(s)$, i.e. b and s cannot match if b arrives before s.

Consider the following greedy algorithm, which attempts to match buyers in their arriving order. An arriving buyer b is matched to the seller with the highest marginal value if the marginal value is positive. If the seller is already matched to another buyer b', b' becomes unmatched and never gets matched again. Formally:

Algorithm 2 Greedy algorithm (Feldman et al. [55])

```
• Input: constrained bipartite graph, $G_{d,\sigma}$.
• For each arrival $i = 1, \ldots, n$:
    – If $i$ is a seller, initialize $p(i) = 0$, and $m(s) = \emptyset$.
    – If $i$ is a buyer:
        ∗ Set $s \in \text{argmax}_{s' \in S}\{v_{is'} - p(s')\}$.
        ∗ If $v_{is} - p(s) > 0$, set $m(s) = i$ and set $p(s) = v_{is}$.
• When a seller $s$ becomes critical: match it to $b = m(s)$ if $m(s) \neq \emptyset$.
```

Proposition 7 (Feldman et al. [55]). *The greedy algorithm is 0.5-competitive for online bipartite constrained graphs.*

Feldman et al. [55] prove that this algorithm is 0.5-competitive for an online matching problem with *free disposal*. In their setting all seller exists and buyer arrive one at a time. The algorithm provides the same guarantees for constrained bipartite graph since, by construction, there is no harm in assuming that all sellers exist rather than arriving over time. The key behind the proof is that the value $p(s)$ function for each seller s is submodular. In fact the result is a special case of a result by Lehmann et al. [75], who study combinatorial auctions with submodular valuations.

6.3.2 An algorithm for arbitrary graphs

In this section we extend the greedy algorithm for constrained bipartite graphs to arbitrary graphs. A naive way to generate a online constrained bipartite graph from an arbitrary one is to randomly assign each vertex to be either a seller or a buyer, independently and with probability $1/2$. Then only keep the edges between each buyer and all the sellers who arrived before her. Formally:

Algorithm 3 Naive Greedy

```
• Input: an online graph with deadline $d$, $G_{d,\sigma}$.
• For each vertex $t = 1, \dots, n$:
      Toss a fair coin to decide whether $i$ is a *seller* or a *buyer*. Construct the online
      constrained bipartite graph $\tilde{G}(d, \sigma)$ by keeping only the edges between each buyer
      and the sellers who arrived before her.
• Run the Greedy algorithm on $\tilde{G}(d, \sigma)$.
```

Corollary 4. *The naive greedy algorithm is $1/8$-competitive for arbitrary online graphs.*

Observe that for vertices i, j with $\sigma(i) < \sigma(j)$, edge (i, j) in the original graph remains in the generated constrained bipartite graph with probability $1/4$ (if i is a seller and j is a buyer). We then use proposition 7 to prove that naive greedy is $1/8$-competitive.

One source of inefficiency in the naive greedy algorithm is that the decision whether a vertex becomes a seller or a buyer is done independently at random and without taking the graph structure into consideration. We next introduce the *Postponed Greedy* algorithm that defers these decisions as long as possible in order to construct the constrained bipartite graph more carefully.

When a vertex k arrives, we add two copies of k to a virtual graph: a seller s_k and a buyer b_k. Let S_t and B_t be the set of sellers and buyers at arrival time t. On arrival, seller s_k does not have any edges, and buyer b_k has edges towards any vertex $s_l \in S_k$ with value $v_{l,k}$. Then we run the greedy algorithm with the virtual graph as input. When a vertex k becomes critical, s_k becomes critical in the virtual graph, and we compute its matches generated by greedy.

Both s_k and b_k can be matched in this process. If we were to honor both matches, the outcome would correspond to a 2-matching, in which each vertex has degree at most 2. Now observe that because of the structure of the constrained bipartite graph, this 2-matching does not have any cycles; it is just a collection of disjoint paths. We decompose each path into two disjoint matchings and choose each matching with probability 1/2.

In order to do that, the algorithm must determine, for each original vertex k, whether the virtual buyer b_k or virtual seller s_k will be used in the final matching. We say that k is a *buyer* or *seller* depending on which copy is used. We say that vertex k is *undetermined* when the algorithm has not yet determined which virtual vertex will be used. When an undetermined vertex becomes critical, the algorithm flips a fair coin to decide whether to match according to the buyer or seller copy. This decision is then propagated to the next vertex in the 2-matching: if k is a *seller* then the next vertex will be a *buyer* and vice-versa. That ensures that assignments are correlated and saves a factor 2 compared to uncorrelated assignments in the naive greedy algorithm.

Algorithm 4 Postponed Greedy (PG)

```
• Input: an online graph with deadline $d$, $G_{d,\sigma}$.
• Process events at time $t$ in the following way:
    1. Arrival of a vertex $k$:
        (a) Set $k$'s status to be undetermined.
        (b) Add a virtual seller: $S_t \leftarrow S_{t-1} \cup \{s_k\}$, $p(s_k) \leftarrow 0$ and $m(s_k) = \emptyset$.
        (c) Add a virtual buyer: $B_t \leftarrow B_{t-1} \cup \{b_k\}$.
        (d) Find a virtual seller for the virtual seller: $s = \text{argmax}_{s' \in S_t} v_{s',b_k} - p(s')$.
        (e) Match if marginal utility is positive: If $v_{s,b_k} - p(s) > 0$, then tentatively match
            $b_k$ to $s$ by setting $m(s) \leftarrow b_k$ and $p(s) \leftarrow v_{s,b_k}$.
    2. Vertex $k$ becomes critical:
        (a) Proceed if no match found: If $m(s_k) = \emptyset$, do nothing.
        (b) match in the virtual graph: If $m(s_k) = b_l$. Set $S_t \leftarrow S_t \setminus \{s_k\}$, and $B_t \leftarrow$
            $B_t \setminus \{b_l\}$.
        (c) If $k$'s status is undetermined, w.p $1/2$ set it to be either seller or buyer.
            i. If $k$ is a seller: finalize the matching of $k$ to $l$ and collect the reward $v_{k,l}$.
               Set $l$'s status to be a buyer.
            ii. If $k$ is a buyer: Set $l$'s status to be a seller.
```

Theorem 6. *The postponed greedy (PG) algorithm is* 1/4*-competitive for arbitrary online graphs.*

Proof. Fix a vertex k, and denote $p^f(s_k)$ to be the final value of its virtual seller s_k's match. If k's status is a seller in step (2.c.i), then we collect $p^f(s_k)$. Note that this happens with probability exactly 1/2 for every k.

$$\text{PG } = \mathbb{E}\left[\sum_{k \text{ is a seller}} p^f(s_k)\right] = \frac{1}{2}\sum_{k\in[1,T]} p^f(s_k).$$

For a virtual buyer b arriving at time t, let $q(b) = \max_{s\in S_t} v_{sb} - p(s)$ be the margin for b in step (1.d). Note that every increase in a virtual seller's price corresponds to a virtual buyer's margin. Using the notation $S = \cup_t S_t$ and $B = \cup_t B_t$, this implies that $\sum_{s\in S} p^f(s) = \sum_{b\in B} q(b)$.

The dual of the offline matching problem linear programs can be written as:

$$\begin{array}{lrll} \text{minimize} & \displaystyle\sum_{k\in[1,T]} \lambda_k & & \\ \text{subject to} & v_{kl} & \leq \lambda_k + \lambda_l \quad \forall (k,l) \text{ s.t. } |k-l| \leq d & \\ & \lambda_k & \geq 0. & \end{array} \tag{Offline Dual}$$

Let i and $j > i$ be two vertices with $j - i \leq d$. When j arrives, we have $q(b_j) \geq v_{ij} - p(s_i)$. Together with the fact that $p(s)$ increases over time, this implies that $\{p^f(s_k) + q(b_k)\}_{k\in[1,T]}$ is a feasible solution to (Offline Dual).

We can conclude that OFF $\leq \sum_k p^f(s_k) + q(b_k) = 2\sum_k p^f(s_k) = 4\text{PG}$.

□

6.3.3 Lower bounds

Claim 1. *When the input is a constrained bipartite graph:*

- *No deterministic algorithm can obtain a competitive ratio above* $\frac{\sqrt{5}-1}{2} \approx 0.618$.

- *No randomized algorithm can obtain a competitive ratio above* $\frac{4}{5}$.

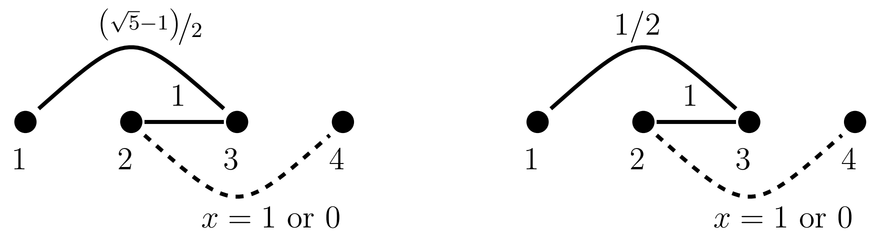

Figure 6-2: Bipartite graph where $S = \{1, 2\}$ and $B = \{3, 4\}$, with $d = 2$: vertex 1 becomes critical before 4 arrives. The adversary is allowed to choose edge $(2, 4)$ to be either 1 or 0. Left: instance for the deterministic case. Right: instance for the randomized case.

Proof. **Deterministic algorithm:** Consider the example on the left of Figure 6-2. When seller 1 becomes critical, the algorithm either matches her with buyer 3, or lets 1 departs unmatched. The adversary then chooses x accordingly. Thus the competitive ratio cannot exceed:

$$\max\left(\min_{x\in\{0,1\}} \frac{\frac{\sqrt{5}-1}{2}+x}{\max(\frac{\sqrt{5}-1}{2}+x,1)}, \min_{x\in\{0,1\}} \frac{1}{\max(\frac{\sqrt{5}-1}{2}+x,1)}\right) = \frac{\sqrt{5}-1}{2}.$$

Randomized algorithm: Consider the example on the right of Figure 6-2. Similarly to the deterministic case, when seller 1 becomes critical, the algorithm decides to match her with 3 with probability p. The adversary then chooses x accordingly. Thus the competitive ratio cannot exceed:

$$\max_{p\in[0,1]} \min_{x\in\{0,1\}} \frac{p(1/2+x)+(1-p)}{\max(1/2+x,1)} = 4/5.$$

□

Next we show that our analysis for PG is tight.

Claim 2. *There exists a* constrained *bipartite graph for which PG is* $^1/_{(4-2\epsilon)}$ *-competitive.*

Proof. Consider the input graph in Figure 6-3. Vertex 2 will be temporarily matched with 3, and vertex 1 will depart unmatched. When 2 becomes critical, with probability $^1/_2$, she

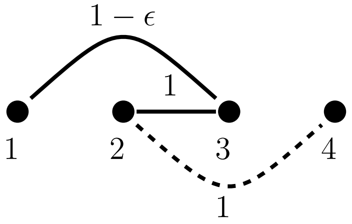

Figure 6-3: Bipartite graph where $S = \{1, 2\}$ and $B = \{3, 4\}$, with $d = 2$: vertex 1 becomes critical before 4 arrives. Dotted edges represent edges that are not know to the algorithm initially.

will be determined to be a *buyer* and will depart unmatched. Therefore the PG collects in expectation $1/2$ while the offline algorithm collects $2 - \epsilon$.

□

6.4 Extensions

6.4.1 Stochastic departures in the adversarial order setting

We relax the assumption that all vertices depart after exactly d time steps.

We therefore focus on the stochastic case, in which the departure time d_i of vertex i is sampled independently from a distribution $\mathcal{D}$. We assume that the realizations d_i are only known at the time i becomes critical.

Proposition 8. *Suppose that there exists $\alpha \in (0, 1)$ such that $\mathcal{D}$ satisfies the property that for all $i < j$,*

$$\mathbb{P}[i + d_i \leq j + d_j | i + d_i \geq j] \geq \alpha.$$

Then PG is $\alpha/4$-competitive.

Proof. When a vertex k becomes critical in the original graph, we match it if its status is determined to be *seller*. In that case, we need to ensure that its tentative match b_l is still present. With probability at least α, vertex l is still present, and we collect $p(s_k)$. The rest of the proof follows similarly to that of Theorem 6 □

Corollary 5. *PG is $1/8$-competitive when $\mathcal{D}$ is memoryless.*

6.4.2 Dynamic deferred acceptance (DDA) algorithm

In this section we provide a new algorithm, called *Dynamic Deferred Acceptance.* Although the DDA provides the same theoretical guarantees as greedy, we present it here since it may lead to better results in practice. Loosely speaking it rationalizes a reoptimization-like algorithm by keeping a tentative maximum weighted matching.

Observe that the greedy algorithm discards a buyer that becomes unmatched and therefore does not attempt to rematch it. We introduce the Dynamic Deferred Acceptance (DDA) algorithm, which takes as input a constrained bipartite graph and returns a matching (formally presented below). The main idea is to maintain a tentative maximum-weight matching m at all times during the run of the algorithm. This tentative matching is updated according to an auction mechanism: at each time t, every seller s is associated with a *price* $p_s(t)$, which is initiated at zero upon arrival. Similarly, every buyer b that that already arrived and yet to become critical is associated with a *profit margin* $q_b(t)$ which corresponds to the value of matching to their most preferred seller minus the price associated with that seller. Every time a new buyer arrives, she bids on her most preferred seller at the current set of prices. This triggers a bidding (or auction) process that terminates when no unmatched buyer can profitably bid on a seller. The auction phase is always initiated at the existing prices and profit margins. This, together with the fact that the graph is bipartite, ensures that prices never decrease and and marginal profits never increase throughout the algorithm. Furthermore, the prices and marginal profits of the sellers and buyers that are present in the "market" form an optimum dual for the matching linear program (see Appendix C.1 for more details).

A tentative match between a buyer and a seller is realized (and the buyer and seller leave) only once the seller becomes critical, i.e., she has been present for d time periods and is about to become critical. At that time, the seller and the buyer are considered matched and depart. This ensures that sellers never get matched before they become critical. A buyer is discarded only if she is unmatched and becomes critical.

Formally, Algorithm 5 provides the procedure for *DDA*. At any point t throughout the algorithm, we maintain a set of sellers S_t, a set of buyers B_t, as well as a matching m, a price $p_s(t)$ for every seller $s \in S_t$, and a marginal profit $q_b(t)$ for every buyer $b \in B_t$.

Algorithm 5 Dynamic Deferred Acceptance

```
• Input: an online graph with deadline $d$, $G_{d,\sigma}$.
• Process each event in the following way:
    1. Arrival of a seller $s$: Initialize $p_s \leftarrow 0$ and $m(s) \leftarrow \emptyset$.
    2. Arrival of a buyer $b$: Start the following ascending auction.
       Repeat
       (a) Let $q_b \leftarrow \max_{s' \in S_t} v_{s',b} - p_{s'}$ and $s \leftarrow \text{argmax}_{s' \in S_t} v_{s',b} - p_{s'}$.
       (b) If $q_b > 0$ then
             i. $p_s \leftarrow p_s + \epsilon$.
            ii. $m(s) \leftarrow b$ (tentatively match $s$ to $b$)
           iii. Set $b$ to $\emptyset$ if $s$ was not matched before. Otherwise, let $b$ be the previous
                match of $s$.
       Until $q_b \leq 0$ or $b = \emptyset$.
    3. Departure of a seller $s$: If seller $s$ becomes critical and $m(s) \neq \emptyset$, finalize the
       matching of $s$ and $m(s)$ and collect the reward of $v_{s,m(s)}$.
```

The ascending auction phase in our algorithm is similar to the auction algorithm by [26]. Prices (for overdemanded sellers) in this auction increase by ϵ to ensure termination, and optimality is proven through ϵ-complementary slackness conditions. For the simplicity of exposition we presented the auction algorithm but for the analysis, we consider the limit $\epsilon \to 0$ and assume the auction phase terminates with the maximum weight matching. Another way to update the matching is through the Hungarian algorithm [74], where prices are increased simultaneously along an alternating path that only uses edges for which the dual constraint is tight.

Lemma 4. *Consider the DDA algorithm on a constrained bipartite graph.*

1. *Throughout the algorithm, prices corresponding the sellers never decrease and the profit margins of buyers never increase.*

2. *At the end of every ascending auction, prices of the sellers and the marginal profits of the buyers form an optimal solution to the dual of the matching linear program associated with buyers and sellers present at that particular time.*

Maintaining a maximum-weight matching along with optimum dual variables does not guarantee an efficient matching for the whole graph. The dual values are not always feasible for the offline problem. Indeed, the profit margin of some buyer b may decrease after some seller departs the market. This is because b may face increasing competition from new buyers, while the bidding process excludes sellers that have already departed the market (whether matched or not).

Proposition 9. *DDA is $1/2$-competitive for constrained bipartite graphs.*

Proof sketch. The proof relies on a primal-dual argument. For any arriving buyer b we denote by q_b^i her *initial profit margin*, i.e. her margin after the ascending auction terminates, and before any other vertex arrives. When a buyer b is matched or departs, we set q_b^f to be her *final profit margin* at that time. Similarly when a seller s departs or is matched, we denote by p_s^f her *final price* at that time.

The proof of the proposition relies on the following three ingredients. First, letting $S = \cup_{t\in[1,T]} S_t$ and $B = \cup_{t\in[1,T]} B_t$, then the algorithm collects

$$\sum_{s\in S} v_{s,m(s)} = \sum_{s\in S} p_s^f + \sum_{b\in B} q_b^f .$$

Second, although the final dual variables (p^f, q^f) are not dual feasible, the pair (p^f, q^i) is a feasible dual solution of the offline matching problem. Finally, we obtain a factor 2 by observing that

$$\sum_{s\in S} p_s^f + \sum_{b\in B} q_b^f = \sum_{b\in B} q_b^i .$$

□

6.4.3 Proof of Proposition 9

First, we observe that by complementary slackness, any seller s (buyer b) that departs unmatched has a final price $p_s^f = 0$ (final profit margin $q_b^f = 0$). When a seller s is critical and matches to b, we have $v_{s,b} = p_s^f + q_b^f$. Therefore, *DDA* collects a reward of $\mathcal{A} = \sum_{s \in S} p_s^f + \sum_{b \in B} q_b^f$.

Second, let us consider a buyer b and a seller $s \in [b-d, b)$ who has arrived before b but not more than d steps before. Because sellers do not finalize their matching before they are critical, we know that $s \in S_b$. An ascending auction may be triggered at the time of b's arrival, after which we have: $v_{s,b} \leq p_s(b) + q_b(b) \leq p_s^f + q_b^i$, where the second inequality follows from the definition that $q_b(b) = q_b^i$ and from the monotonicity of sellers' prices (Lemma 4). Thus, (p^f, q^i) is a feasible solution to the offline dual problem.

Finally, we observe that upon the arrival of a new buyer, the ascending auction does not change the sum of prices and margins for vertices who were already present:

Claim 3. *Let b be a new buyer in the market, and let p, q be the prices and margins before b arrived, and let S_t and B_t be the sets of sellers and buyers present before b arrived. Let p', q' be the prices and margins at the end of the ascending auction phase. Then:*

$$\sum_{s \in S_t} p_s + \sum_{b \in B_t} q_b = \sum_{s \in S_t} p'_s + \sum_{b \in B_t} q'_b. \tag{6.1}$$

The proof of Claim 3 is deferred to Appendix C.1. By applying this equality iteratively after each arrival, we can relate the initial margins q^i to the final margins q^f and prices p^f:

Claim 4. $\sum_{s \in S} p_s^f + \sum_{b \in B} q_b^f = \sum_{b \in B} q_b^i$.

This completes the proof of Proposition 9 given that the offline algorithm achieves at most:

$$\mathcal{O} \leq \sum_{s \in S} p_s^f + \sum_{b \in B} q_b^i \leq 2\mathcal{A}.$$

6.5 Next steps: beating $1/4$

In this section, we discuss potential options to improve on the $1/4$ factor. One possibility is to design algorithms directly for general graphs without using the reduction to bipartite constrained graphs. While we think this is a promising direction, we do not

Another option is to improve upon the factor $1/2$ for bipartite constrained graphs. Two strategies come to mind: First we can study algorithms which also work in the *Free Disposal* setting where decisions need to be finalized upon arrival of a buyer. This is the subject of active research, such as [132]. Second, we can use the fact that our deadline model allows for more flexibility: as long as a vertex is present, we may change her matches. This is the key idea in the DDA algorithm, where we postpone the decision of whom to match until the last possible minute. While this doesn't improve the worst case bound beyond $1/2$, we see in simulations (see Chapter 7) that this idea performs well on practical settings.

6.5.1 α-DDA algorithm

One drawback of the *DDA* algorithm is that it is not greedy enough: it may choose to let a seller depart unmatched when it is not included in a maximum weight matching, even though including this seller is "almost" a max weight matching. This is the case, for example, with the example in Figure 6-3.

In this section, we introduce a variation, called α-DDA which we conjecture outperforms *DDA* both in the worst case and in practical settings.

At a high level, the algorithm is inspired from the *DDA* algorithm: we maintain a maximum weight matching over the vertices that are present. The key difference is that when a seller is about to depart, we increase all of its edges by a factor $\alpha \geq 1$, and re-compute the maximum-weight matching. We can interpret this new algorithm as an interpolation between *DDA* and greedy: when $\alpha = 1$, this is exactly *DDA*, and when $\alpha \to \infty$, this is the greedy algorithm. In other words, the larger α is, and the more we are willing to accept a sub-optimal match for the departing seller. Note that when $\alpha \geq 2$, it's easy to construct an example from the one in Figure 6-1 that leads to a factor 1/2. Thus we can assume $\alpha \in [1, 2)$.[1]

We now detail how to perform the step to re-compute a matching after increasing the edges adjacent to the critical vertex. Let s be the critical seller at time t, and let $p_s(t)$ be her price at that time. Let $N(s)$ be the set of all buyers that are neighbours of s, and set $\tilde{p}_s(t) = \max_{b \in N(s)} \alpha v_{s,b} - q_b(t)$. Note that using $\tilde{p}_s(t)$ instead of $p_s(t)$ restores dual feasibility. We can now run an " auction" in reverse, which is started the critical seller s now bids on her highest margin neighbour, etc. Similarly to the proof of Claim 6.1, there is conservation of the dual values, except for $\tilde{p}_s(t)$.

Conjecture 1. *α-DDA is $\frac{\sqrt{5}-1}{2}$-competitive on bipartite constrained graphs when $\alpha = \frac{\sqrt{5}-1}{2}$.*

6.5.2 An incomplete primal-dual proof of Conjecture 1.

Claim 5. *For every seller s, $p_s^f \geq \alpha p_s(t)$ where t is the time when s becomes critical.*

Proof. If s was not tentatively matched at t, then by complementary slackness $p_s(t) = 0$ and the result is trivial. We now assume that s was tentatively matched to a buyer b at t. Let us denote $p_s(t^-)$ and $q_b(t^-)$ ($p_s(t^+)$ and $q - b(t^+)$) the margins of b at the begining (respectively end) of the reverse auction step.

Because b is now unmatched at the start of the *reverse auction*, any bid on b will immediately terminate the auction. Therefore $q_b(t^+) = q_b(t^-)$. Because of complementary slackness, we have $p_s(t^-) + q_b(t^-) = v_{s,b}$. We can conclude:

$$p_s^f = p_s(t^+) \geq \alpha v_{s,b} - q_b(t^+) = \alpha p_s(t^-) + \alpha q_b(t^-) - q_b(t^+) \geq \alpha p_s(t^-).$$

□

Remarks

1. Note that Dual feasibility implies that for any time t we have $v_{s,b} \leq p_s(t) + q_b(t)$ if s and b are present in the pool.

[1] This goes contrary to the popular French saying "Un tiens vaut mieux que deux tu l'auras" (a bird in the hand is worth two in the bush).

2. If s becomes critical at t and finally matches to b, the algorithm collects $v_{s,b} = \frac{1}{\alpha}\left(p_s^f + q_b^f\right)$.

Assumption 1. *We assume for now that similarly to the* DDA *algorithm, the dual variables $p_s(t)$ and $q_b(t)$ are monotone functions of t.*

Consider two vertices s, b. Two cases can occur:

Case 1: b is matched before s becomes critical. Then if t is the time at which b is matched, we have: $v_{s,b} \leq p_s(t) + q_b(t)$ and using Claim 5 and Assumption 1 that p_s increases over time, we get $v_{s,b} \leq \frac{1}{\alpha}p_s^f + q_b^f$.

Case 2: b is matched after s becomes critical. Then if t is the time at which s becomes critical, by definition of our *reverse auction* step, we have: $p_s^f \geq \alpha v_{s,b} - q_b(t)$. Therefore, $v_{s,b} \leq \frac{1}{\alpha}\left(p_s^f + q_b^{(i)}\right)$.

Therefore, $\left(\frac{1}{\alpha}p_s^f, \max\left(q_b^f, \frac{1}{\alpha}q_b^{(i)}\right)\right)$ is a dual feasible solution to the offline problem. Denote $\mathcal{O}$ to be the reward collected by the offline algorithm. We have:

$$\alpha\mathcal{O} \leq \sum_s p_s^f + \sum_j \max(\alpha q_b^f, q_b^{(i)}) \leq \sum_s p_s^f + \alpha\sum_{b\in J} q_b^f + \sum_{b\in J^c} q_b^{(i)},$$

Where J is the set of buyers such that $\alpha q_b^f \geq q_b^{(i)}$.

Claim 6. *Similarly to Claim 4 in the case $\alpha = 1$, we have:*

$$\sum_{b\in J^c} q_b^{(i)} \leq \sum_{b\in J^c} q_b^f + \frac{1}{\alpha}\sum_s p_s^f$$

Proof. The proof follows the same induction idea as in Claim 4: if we denote t_s to be the departing time of seller s, then using Claim 5, it is enough to prove that

$$\sum_{b\in J^c} q_b^{(i)} \leq \sum_{b\in J^c} q_b^f + \sum_s p_s(t_s).$$

Under Assumption 1, that buyer's margins are monotone, the duals for buyers in set J^c keep decreasing. Their aggregate decrease is at most the aggregate increase in all of the seller's prices.

□

Using Claims 5 and 6, we can conclude:

$$\mathcal{O} \leq \left(1 + \frac{1}{\alpha}\right) \sum_i p_i^f + \alpha \sum_b q_b^f \leq \max\left(1 + \frac{1}{\alpha}, \alpha\right) \mathcal{A}$$

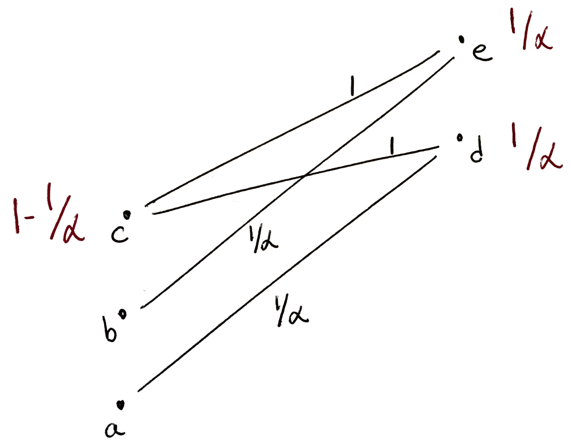

Figure 6-4: In this example, vertices arrive in the order a, b, c, d, e. a,b and c are sellers, and start with prices $p_a = p_b = p_c = 0$. d and e are buyers, when d arrives, its margin is 1 and no auction is started. When e arrives, its margin is initially 1 and starts an auction. At the end of the auction, $p_c = 1 - {}^1\!/\!_\alpha$ and $q_d = q_e = {}^1\!/\!_\alpha$ these values are denoted in red. At the time when a becomes critical, the reverse auction leads to $p_c = 0$ and $q_d = q_e = 1$ which violates monotonicity.

Remark 3. *Two comments are in place:*

- *Note that Assumption 1 is not always verified. In Figure 6-4 we provide an example where buyers' margins increase and sellers' prices decrease during the* reverse auction *phase.*

- *Solutions to correct this shortcoming may fall in two categories. One could use a charging mechanism to account for the non-monotonicity in the duals. Another idea* *would be to use another mechanism to restore dual feasibility after increasing the edges adjacent to the critical vertex, in a way that maintains monotonicity.*

6.6 Conclusion

This paper introduces a model for dynamic matching in which all agents arrive and depart after some deadline. Match values are heterogeneous and the underlying graph is non-bipartite. We study online algorithms for two settings, where vertices arrive in an adversarial or random order.

We introduce two new 1/4-competitive algorithms when departures are deterministic and known in advance. We also provide a 1/8-competitive algorithm when departures are stochastic, i.i.d, memoryless, and known at the time a vertex becomes critical. Finally we show that no online algorithm is more than 1/2-competitive.

Importantly, our model imposes restrictions on the departure process and requires the algorithm to know when vertices become critical. Other than closing the gaps between the upper bound $1/2$ and the achievable competitive ratios, we point out a just a few interesting directions for future research. Our model imposes that matches retain the same value regardless of when they are conducted. An interesting direction is to account for agents' waiting times. A different intersting objective is to achieve both a high total value and a large fraction of matched agents. Finally, it is interesting to consider the stochastic setting with prior infromation over weights and future arrivals.

Chapter 7

Maximum Weight Online Matching with Deadlines in Random Order

7.1 Introduction

In this chapter, we study the setting in which vertices arrive in **random order**. We analyze a *batching* algorithm which, every $d+1$ time steps, computes a maximum weighted matching among the last $d+1$ arrivals. Vertices that are left unmatched are discarded forever. We show that when the number of vertices is sufficiently large, batching is 0.279-competitive.

The analysis proceeds in three steps. First, we show that the competitive ratio is bounded by the solution to a graph covering problem. Second, we show how a solution for small graphs can be extended to covers for larger graphs. Finally, we establish a reduction that allows us to consider only a finite set of values for d. We conclude with a computer-aided argument for graphs in the finite family.

7.2 Random arrival order

In some cases, the vertices can be assumed to come from a distribution that is unknown to the online algorithm. One way to model this is to assume that the adversary chooses the underlying graph, but that the vertices arrive in random order.

7.2.1 The batching algorithm

The *batching* algorithm computes a maximum-weight matching every $d+1$ time steps. Every vertex in the matching is then matched, and all other vertices in the batch are discarded.

Theorem 7. *Batching is* $(0.279 + O(1/n))$*-competitive.*

The proof of Theorem 7 works in three steps. In a first step, we reduce the analysis of the competitive ratio of *Batching* to a graph covering problem. More precisely, we show that it is enough to cover C_n^d, the cycle with n vertices to the power d, with ensembles of cliques. Second, we show how a cover for small n can be extended to any n at the cost of a small rounding error. Finally, we establish a reduction that allows us to consider only a finite set of values for d. We conclude with a computer-aided argument for graphs in the finite family.

Reducing to a graph theoretic problem

There is no harm in assuming that the underlying graph G is a complete. Recall that S_n is the set of all permutations over integers $1, ..., n$. For any deadline d and any arrival sequence $\sigma \in S_n$, we define the *path* graph $P_n^d(\sigma)$ with edge-weight $v_{ij} = 1$ if $|\sigma(i) - \sigma(j)| \leq d$, and $v_{ij} = 0$ otherwise.[1]

Note that every batch in the algorithm has $d+1$ vertices except the last batch which may have fewer vertices. Let $b_i(\sigma, d)$ be the batch of vertex i under permutation σ and batch size $d+1$: $b_i(\sigma, d)$ is the unique integer such that $(d+1)(b_i - 1) < \sigma(i) \leq (d+1)b_i$. We define the *batched* graph $B_n^d(\sigma)$ with edge-weight $v_{ij} = 1$ if i and j are in the same batch (i.e. $b_i(\sigma, d+1) = b_j(\sigma, d+1)$), and $v_{ij} = 0$ otherwise.[2]

For any $n \geq d \geq 1$, denote C_n^d to be the n-cycle to the power d.

Definition 5 (Graph operations). *For any two graphs H and H' with vertices $1, ..., n$ and respective edge weights v_{ij}, v'_{ij}, we define the following:*

(i) The linear combination $aH + bH'$ denotes the graph with edge weights $av_{ij} + bv'_{ij}$,

(ii) The product $H * H'$ denotes the graph with edge weights $v_{ij} * v'_{ij}$, and

(iii) We say that H is a cover of H' if for all i, j, $v_{i,j} \geq v'_{ij}$.

For any graph H, let $m(H)$ denote the value of a maximum-weight matching over H. Observe that when the arrival sequence is σ, the graph $P_n^d(\sigma) * G = G(d, \sigma)$ and therefore the offline algorithm collects $m(P_n^d(\sigma) * G)$. Note that the online algorithm collects $m(B_n^d(\sigma) * G)$.

Remark 4. *Observe that for any graphs H, H', G and any $a, b \in \mathbb{R}$, we have:*

- *$m(aH + bH') \leq am(H) + bm(H')$.*

- *If H is a cover of H', then $m(H * G) \geq m(H' * G)$.*

[1] Note that $P_n^d(\sigma)$ corresponds to the path $(\sigma(1), \sigma(2)), (\sigma(2), \sigma(3)), ..., (\sigma(n-1), \sigma(n))$ taken to the power d.

[2] Note that $B_n^d(\sigma)$ is a collection of disjoint $(d+1)$-cliques.

Definition 6 (Periodic permutation)**.** *For $p < n$ such that p divides n, we say that a permutation $\sigma \in S_n$ is p-periodic if for all $i \in [1, n-p]$, $\sigma(i+p) \equiv \sigma(i) + p \mod n$.*

We say that a permutation σ is periodic if there exists p such that σ is p-periodic.

Definition 7 ((α, d)-cover)**.** *Let F be an unweighted graph with n vertices. We say that a set of permutations $\{\sigma_1, ..., \sigma_K\} \in S_n$ forms an (α, d)-cover of F if there exist values $\lambda_1, ..., \lambda_K \in [0,1]$ such that:*

(i) $\sum_{k \leq K} \lambda_k B_n^d(\sigma_k)$ *is a cover of F.*

(ii) $\sum_{k \leq K} \lambda_k = \alpha$.

We say that an (α, d)-cover is p-periodic if for all k, σ_k is p-periodic.

The next proposition will allow us to abstract away from the weights that are chosen by the adversary. For any graph H, we denote by H_{ij} the weight v_{ij} in H.

Proposition 10. *If there exists an (α, d)-cover of C_n^d, then batching is $1/\alpha$-competitive.*

Proof. Let id be the identity permutation over n vertices. Let $\{\sigma_1, ..., \sigma_K\}$ be an (α, d)-cover of C_n^d. Fix an arrival sequence $\sigma \in S_n$. We first claim that $\{\sigma_1 \circ \sigma, ..., \sigma_K \circ \sigma\}$ is an (α, d)-cover of $P_n^d(\sigma)$.

For any $\sigma \in S_n$, let us denote $\beta_{i,j}(\sigma)$ and $\rho_{i,j}(\sigma)$ to be the weights of edge (i, j) in $B_n^d(\sigma)$ and $P_n^d(\sigma)$ respectively. Consider $(i, j) \in P_n^d(\sigma)$: $|\sigma(i) - \sigma(j)| \leq d$:

$$\begin{aligned}
\sum_k \lambda_k \beta_{i,j}(\sigma_k \circ \sigma) &= \sum_k \lambda_k \mathbb{I}[b_i(\sigma_k \circ \sigma, d) = b_j(\sigma_k \circ \sigma, d)] \\
&= \sum_k \lambda_k \mathbb{I}[b_{\sigma(i)}(\sigma_k, d) = b_{\sigma(j)}(\sigma_k, d)] \\
&\geq \rho(\mathrm{id})_{\sigma(i), \sigma(j)} = 1,
\end{aligned}$$

where the last inequality is implied by the fact that $\{\sigma_1, ..., \sigma_K\}$ is an (α, d)-cover of C_n^d and therefore of $P_n^d(\mathrm{id})$. Therefore the claim holds using remark 4.

Denote by BAT the value collected by the batching algorithm and OFF the value collected by the offline algorithm. Observe that

$$\begin{aligned}
\mathrm{OFF} &= \frac{1}{n!}\sum_{\sigma\in S_n} m(P_n^d(\sigma) * G)\\
&\leq \frac{1}{n!}\sum_{\sigma\in S_n}\sum_{k}\lambda_k m(B_n^d(\sigma_k\circ\sigma) * G)\\
&= \frac{1}{n!}\sum_{k}\lambda_k\sum_{\sigma'\in S_n} m(B_n^d(\sigma') * G)\\
&= \alpha\mathrm{BAT},
\end{aligned}$$

where we used the change of variable $\sigma' = \sigma_k \circ \sigma$ and the fact that the application $\mathcal{A}_k : \sigma \mapsto \sigma_k \circ \sigma$ is a bijection.

□

We have reduced the analysis of Batching to a graph-theoretic problem without edge weights. In what follows, we will show that we can reduce the problem further to find covers of C_n^d for only small values of n and d.

Reducing n: periodic covers.

We now wish to find (α, d)-covers for C_n^d for every n and d. In Proposition 11, we show that it is sufficient to find periodic covers for small values of n.

Proposition 11. *Let p be a multiple of $d+1$, and n_1 a multiple of p. Any p-periodic (α, d)-cover of $C_{n_1}^d$ can be extended into an $(\alpha + O(p/n), d)$-cover of C_n^d for any $n \geq n_1$.*

Proof when n is a multiple of p. Let $\{\sigma_1, ..., \sigma_K\}$ be a p-periodic (α, d)-cover of $C_{n_1}^d$. We will show that it can be extended into an (α, d)-cover of C_n^d.

Assume for now that n is a multiple of p. Let σ_k' be the p-periodic permutation over $1, ..., n$ such that for all $i \in [1, p]$, $\sigma_k'(i) = \sigma_k(i)$. Take $i', j' \in [1, n]$ such that $|i' - j'| \leq d$. Because $n_1 > p$ is a multiple of p, there exist $i, j \in [1, n_1]$ such that $i \equiv i' \mod p$, $j \equiv j'$

mod p and $|i-j| \leq d$. By p-periodicity of σ_k and σ'_k, we know that $B_n^d(\sigma'_k)_{i',j'} = B_{n_1}^d(\sigma_k)_{i,j}$. Thus we can conclude that $\{\sigma'_1, ..., \sigma'_K\}$ is an (α, d)-cover of C_n^d. $\square$

In the case when n is not a multiple of p, the proof follows similar ideas and looses an additional factor $\left(\frac{n}{n-p}\right)^2$ due to rounding of n to a lower multiple of p. Details are provided in Appendix D.1.

Reducing d: cycle contraction.

In Proposition 11, we show that it is enough to find periodic (α, d)-covers of C_n^d for small values of n. Next, we provide a reduction that enables us to consider only a finite set of values for d.

The key idea of the reduction is that we can contract vertices of C_n^d into n/u groups of u vertices. The resulting graph also happens to be a cycle $C_{n/u}^{(d+1)/u}$. In Proposition 12, we provide a way to expand an $(\alpha, u-1)$-cover on the contracted graph into an (α, d) cover on the original graph.

Definition 8 (Cycle contraction)**.** *For any n, d and an integer u which divides n, we define the u-contraction $f_u(C_n^d)$ to be the graph with vertices $a_k = \{uk+1, ..., u(k+1)\}$ for $k \in [0, {}^{n}\!/_{u} - 1]$, and edges (a_k, a_l) if and only if there exist $i \in a_k$ and $j \in a_l$ with an edge (i, j) in C_n^d.*

Claim 7. *For any d, if $u > 1$ divides $d+1$ and $d+1$ divides n, then $f_u(C_n^d) = C_{n/u}^{(d+1)/u}$.*

Proof. We first prove that $C_{n/u}^{(d+1)/u}$ covers $f_k(C_n^d)$. Fix $k, l \in [0, {}^{n}\!/_{u} - 1]$, and assume that $k < l$. If $|l-k| \leq (d+1)/u$, then let $i = u(k+1)$ and $j = ul+1$. We have $|j-i| = u(l-k-1)+1 \leq d$, thus $(i, j) \in C_n^d$ and $(k, l) \in f_u(C_n^d)$.

Conversely, we now prove that $f_u(C_n^d)$ covers $C_{n/u}^{(d+1)/u}$. If there exist $i \in a_k$ and $j \in a_l$ such that $|j-i| \leq d$, then $u(l-k) \leq ul + 1 - u(k+1) \leq d+1$ which implies that $(k, l) \in C_{n/u}^{(d+1)/u}$. $\square$

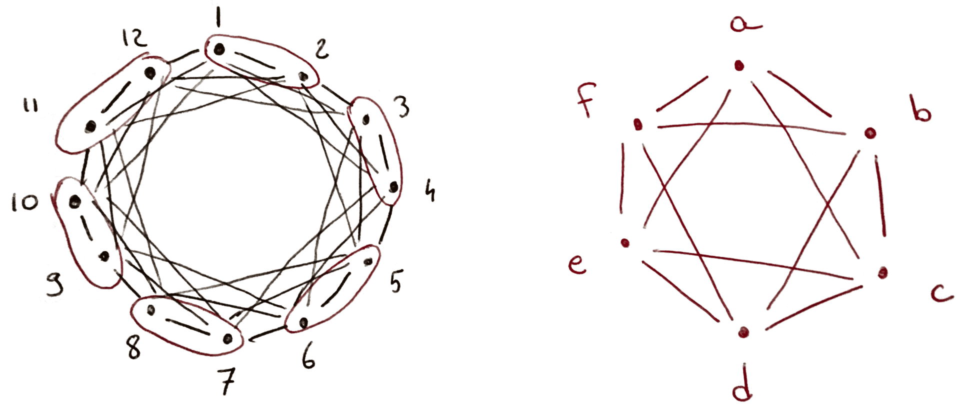

Figure 7-1: Left: C^3_{12}, with contraction for $u = 2$. Right: Contracted graph $f(C^3_{12}) = C^2_6$ with vertices $a = \{1, 2\}$, $b = \{3, 4\}$, ... $f = \{11, 12\}$.

Proposition 12. *Fix $d \geq 1$. For $d + 1 > k \geq 1$, suppose that there is a periodic $(\alpha, k-1)$-cover of C^k_{rk}.*

(i) For any integer r, if k divides $d+1$ then there exists a periodic (α, d)-cover of $C^d_{r(d+1)}$.

(ii) In general, if v is the remainder of the euclidian division of $d+1$ by k, then there exists a periodic $(\alpha(1 + {}^{v}\!/_{d+1-v})^2, d)$-cover of $C^d_{r(d+1)}$.

Proof of (i). Suppose that $d + 1 = ku$ and suppose that there exists p multiple of $d + 1$ such that we have a p-periodic $(\alpha, k - 1)$-cover $\{\sigma_1, ..., \sigma_K\}$ of $f_u\left(C^d_{r(d+1)}\right) = C^k_{rk}$. For any permutation $\sigma \in S_{rk}$ we can construct a permutation $\sigma' \in S_{r(d+1)}$ in the following way: if $i \in a_t$ then $\sigma'(i) = \frac{n}{k}\sigma(t) + i$. Because $B^k_{rk}(\sigma_i)$ is a cover of $B^d_{r(d+1)}$, we can conclude that $\sigma'_1, ..., \sigma'_K$ is an (α, d)-cover of $C^d_{r(d+1)}$.

□

The proof for case (ii) follows a similar idea, with an additional randomization that chooses a subset $d + 1 - v$ vertices that we can group in every group of $d + 1$. The details are in Appendix D.1.

Final step: Computer-aided proof of factor 2.79

We will now apply Proposition 11 with $p = 2(d+1)$ and $n_1 = 4(d+1)$. Let Ω_d be the set of $2(d+1)$-periodic permutations of $1, ..., 4(d+1)$. We can find covers for $C^d_{4(d+1)}$ using the following linear program:

$$\begin{aligned} \text{minimize} \quad & \sum_{\sigma \in \Omega_d} \lambda_\sigma \\ \text{subject to} \quad & \sum_{\sigma \in \Omega_d} \lambda_\sigma \mathbb{I}[b_i(\sigma, d) = b_j(\sigma, d)] \geq 1, \quad \forall (i,j) \in C^d_{4(d+1)} \\ & \lambda_\sigma \in \mathbb{R}^+, \qquad\qquad\qquad\qquad \sigma \in \Omega_d \end{aligned} \tag{LP$_d$}$$

Proposition 13. *Let α_d be the solution to LP_d. Let $\alpha = \sup_{d \geq 1} \alpha_d$. Then* Batching *is $(1/\alpha + O(1/n))$-competitive.*

Proof. Follows from Propositions 10 and 11. □

The Linear program (LP$_d$) has $O(d!)$ variables, and solving it may not be computationally possible when d is large. Using Proposition 12, we now provide a way to find upper bounds on α_d by solving a different LP on a smaller graph.

Recall that Ω_{k-1} is the set of $2k$-periodic permutations of $1, ..., 4k$. We define the problem of finding an $(\alpha, k-1)$-cover of the cycle C^k_{4k}.

$$\begin{aligned} \text{minimize} \quad & \sum_{\sigma \in \Omega_{k-1}} \lambda_\sigma \\ \text{subject to} \quad & \sum_{\sigma \in \Omega_{k-1}} \lambda_\sigma \mathbb{I}[b_i(\sigma, k-1) = b_j(\sigma, k-1)] \geq 1, \quad \forall (i,j) \in C^k_{4k} \\ & \lambda_\sigma \in \mathbb{R}^+, \qquad\qquad\qquad\qquad \sigma \in \Omega_{k-1} \end{aligned} \tag{LP'$_k$}$$

We denote by α'_k the solution to (LP'$_k$). Solving (LP'$_k$) numerically for $k = 4$ yields $\alpha'_4 \leq 3.17$. For all $d \geq 52$ Proposition 12 therefore implies that, $\alpha_d \leq 3.17 * \left(\frac{51}{49}\right)^2 = 3.58$. [3]

[3] We note that our methodology can be extended to obtain a better factor. For instance, being able to solve (LP$_d$) for values higher than 50 would lead to a competitive ratio closer to $\frac{1}{3}$.

For $d \leq 50$, we either solve (LP$_d$) directly, or use Proposition 12 to show that $\alpha_d \leq 3.58$ (see Appendix D.2). Observing that $2.79 \leq \frac{1}{3.58}$, this concludes the proof for Theorem 7. □

7.2.2 Lower bound in random order.

Proposition 14. *No algorithm is more than $\frac{1}{2}$-competitive even under the random arrival order.*

Proof. Consider a graph with three vertices $\{1, 2, 3\}$ and $d = 1$, i.e. vertices can only be matched to the ones arriving just before or after them. After the first two arrivals, the online algorithm $\mathcal{A}$ needs to decide whether to match them or let the first arrival leave. Furthermore, it has no information on how $v_{\sigma(1),\sigma(2)}$ compares to the other edge weights. Therefore the decision of whether to match has to be a coin toss. Regardless of whether the algorithm matches the first two or the last two arrivals, its expected reward is $\frac{v_{1,2}+v_{2,3}+v_{3,1}}{3}$. OFF however has an expected reward of $\frac{\max(v_{1,2},v_{2,3})+\max(v_{2,3},v_{3,1})+\max(v_{3,1},v_{1,2})}{3}$. Taking $v_{1,2} = v_{2,3} \to 0$, we get $\mathcal{A} = 1/3$ while OFF $= 2/3$ which concludes the proof. □

7.3 Extension: the value of look-ahead information

We assume now that the online algorithm has advance knowledge of the vertices that will arrive in l time steps (and their adjacent edges). We can update the Batching Algorithm in the following way: every $d+l+1$ time steps, compute a maximum-weight matching on both the current vertices and the next l arrivals. Match vertices as they become critical according to the matching, and discard unmatched vertices. Note that this is the same as running Batching when the deadline is $d + l$.

Proposition 15 shows that having information about even d future arrivals can lead to increases beyond the upper limit of 1/2 on the improvements through better algorithms.

Proposition 15. *There exists an $(\frac{d+l+1}{l+1}, d+l)$-cover of C_n^d.*

Proof. For $k \in [0, d+l]$, let $\sigma_k(i) = i + k \mod n$. Let i, j be such that $|i - j| \leq d$, then

$b_i(\sigma_k, d) = b_j(\sigma_k, d)$ for at least $l+1$ different values of k. We can conclude that $\sigma_0, ..., \sigma_{d+l}$ is a $(\frac{d+l+1}{l+1}, d+l)$-cover by taking $\lambda_0 = ... = \lambda_{d+l} = \frac{1}{l+1}$. □

Corollary 6. *Batching with l-lookahead is $\frac{l+1}{d+l+1}$-competitive when n is large.*

7.4 Numerical results

In this section, we study the performance of different algorithms on graphs generated using two real-world datasets. We are interested in comparing the *DDA* algorithm, introduced in Chapter 6 to the *Batching* algorithm, which we studied in this chapter.

Recall that *DDA* randomizes vertices into two groups, and maintains a maximum-weight matching over resulting bipartite graph through re-optimization. Vertices are only matched when they become *critical*, i.e. when they are about to depart from the graph. In this section, we consider a simpler, non-randomized, variant termed *Re-Opt*, in which a maximum-weight matching is computed over the whole graph (and not a bipartite subset). While this simplified algorithm does not have theoretical guarantees, we are able to show that it performs well, both when departures are deterministic and geometrically distributed.

We will also consider a variant inspired from *α-DDA*, termed *α-Re-Opt*, which assigns a multiplicative weight α to vertices which are about to depart, as a way to biais the maximum-weight matching towards vertices which are about to leave.

7.4.1 Setup

Data

The first experiment uses data from the National Kidney Registry (NKR), which consists of 1681 patient-donor pairs who have joined the NKR. For any two patient-donor pairs k and t, the data allows to verify whether they could have formed a 2-way exchange (match) if they had been present at the same time. If that is the case, we set $v_{k,l} = 1$, otherwise and

$v_{k,l} = 0$ (in particular we simply try to maximize the number of matches).[4]

In the second instance, we use New York City yellow cabs dataset [5], which consists of rides taken in NYC over a year. For any pair k, l of trips, we can compute the Euclidian distance that would have been traveled had the two passengers taken the same taxi (with multiple stops). The value $v_{k,l}$ represents the "distance saved" by combining the trips.

Numerical Simulations

In both cases, this enables us to generate a dynamic graph in the following way. For $t \in [1, T]$:

1. Sample with replacement a pair t from the dataset.

2. For any pair l that is present in the pool, compute the value $v_{t,l}$ of matching t to l.

3. Sample a departure time $d_t \sim \mathcal{D}$.

We consider three departure settings, termed *deterministic*, *exponential* and *uniform* respectively, in which $\mathcal{D}$ is either constant with value d, or exponentially distributed with mean d. We will report simulation results for $d = 50, 75, 100, 125, 150$.

Some of the algorithms in Section 7.4.2 are parameterized. In that case, we only plot the out of sample performance when selecting the parameter on a validation subset of the data.

7.4.2 Results

Value of knowing Departure times

In this section, we assume that *critical* vertices reveal to the online algorithm that they are about to depart. We will say that the departures are *known*. Note that in particular in the case of *exponential* and *uniform* departures, the online algorithm has access to both the distribution and the realization of the departure times for each vertex.

[4] We ignore here the possibility of larger exchanges between 3 pairs or chains, which are common in practice.

[5] http://www.andresmh.com/nyctaxitrips/

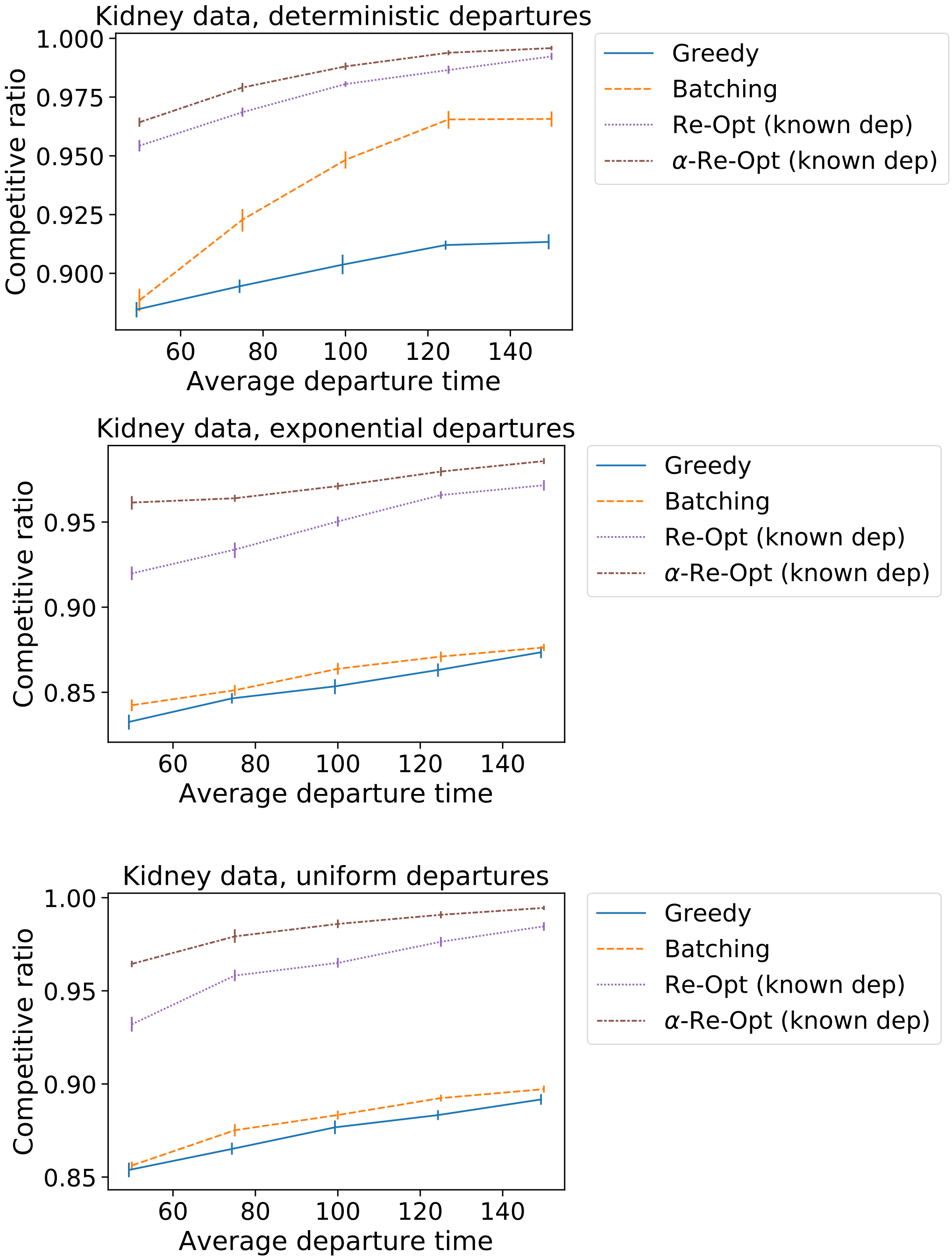

Figure 7-2: Performance of *Greedy*, *Batching*, *Re-Opt* and α-*Re-Opt* on kidney data. The compatibility graph G is binary-weighted.

In Figures 7-2 and 7-3, we show that the *Re-Opt* algorithm is able to effectively use this information in order to significantly increase the number of matches when compared to both *Batching* and *Greedy.*

In Figure 7-2, we observe that *Batching* is significantly outperforming *Greedy* only in the case where departures are deterministic. The intuition is that when departures are stochastic, *Batching* runs the risk that some vertices are never considered in a match run. When departures are deterministic however, as long as $b \leq d$, all vertices will be included in at least one match run.

In Figure 7-3, we note that greedy performs significantly worse than in the case of the kidney dataset (Figure 7-2) which had a binary-weighted compatibility graph. Observe that both *Re-Opt* and *Batching* algorithms outperform *Greedy.* The intuition here is that because edges can take any non-negative value, the value of making some agents wait is now higher. We also observe that the benefit of α-*Re-Opt* over *Re-Opt* are much more pronounced in the case of the kidney simulations than in the case of taxi data.

Re-Optimization *vs* Batching

In our implementation of *Re-Opt* in the previous section, we assumed that the online algorithm knows when vertices became critical. In other words, we had access to the departure *realization.* This additional information led to an increase in the number of matches, when compared to *Batching.*

In this section, we implemented *Re-Opt* in a way that doesn't use the information of vertex *criticality.* To do this, we introduce a *patience* parameter p, which acts similarly to a batch size. Instead of waiting for vertices to become critical, we match an edge whenever at least one of the vertices has been waiting for more than the p time steps. As with the batching parameter b, we only show out of sample results for the best value of p.

In Figure 7-4, we see that in the case of deterministic departures, the critical times are de facto known, and results are similar to those of section 7.4.2. In the case of either *exponential* or *uniform* departures, *Re-Opt* and *Batching* seem to be performing similarly.

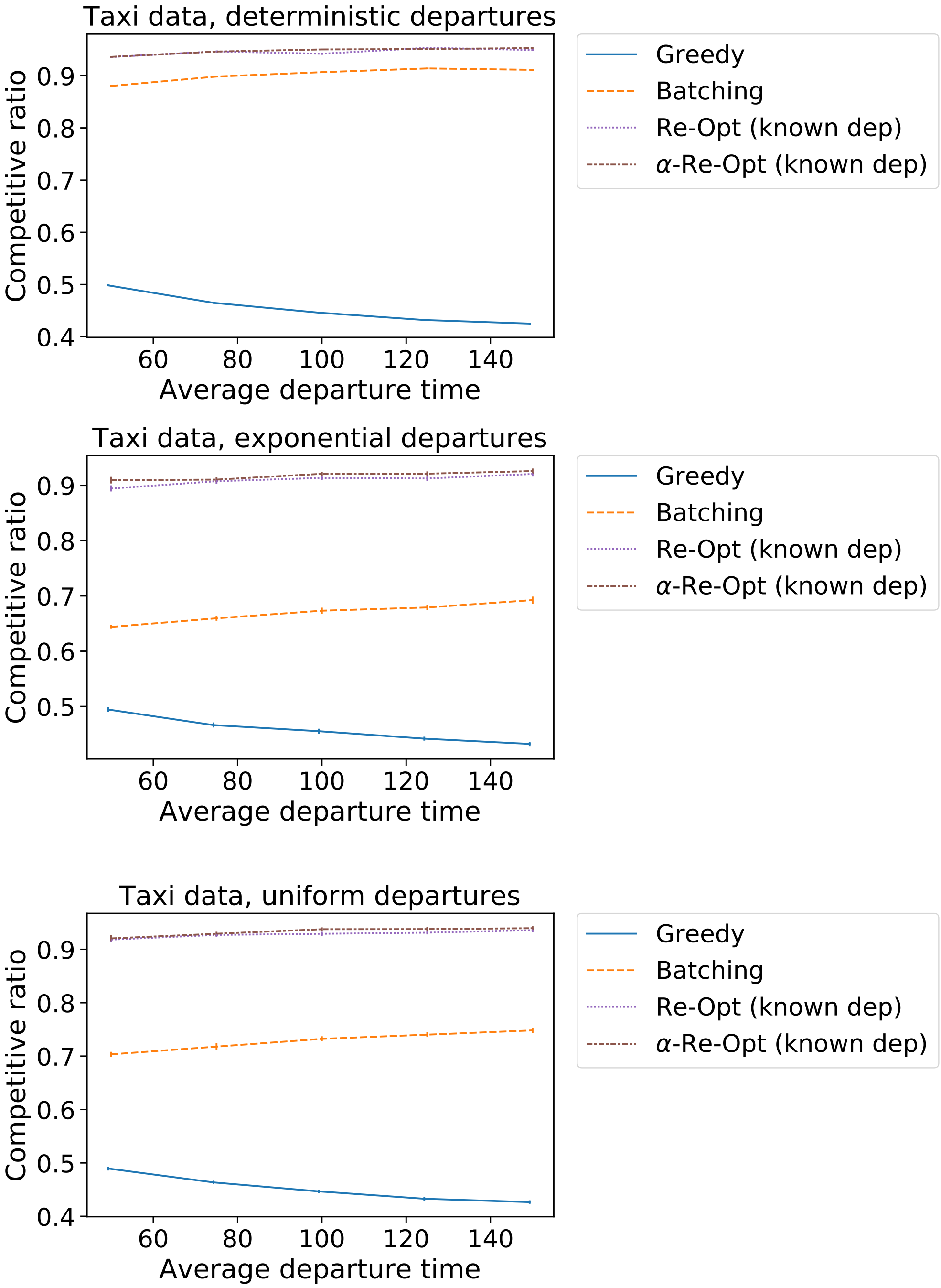

Figure 7-3: Performance of *Greedy*, *Batching*, *Re-Opt* and α-*Re-Opt* on taxi data. The compatibility graph G is weighted $\in \mathbb{R}$.

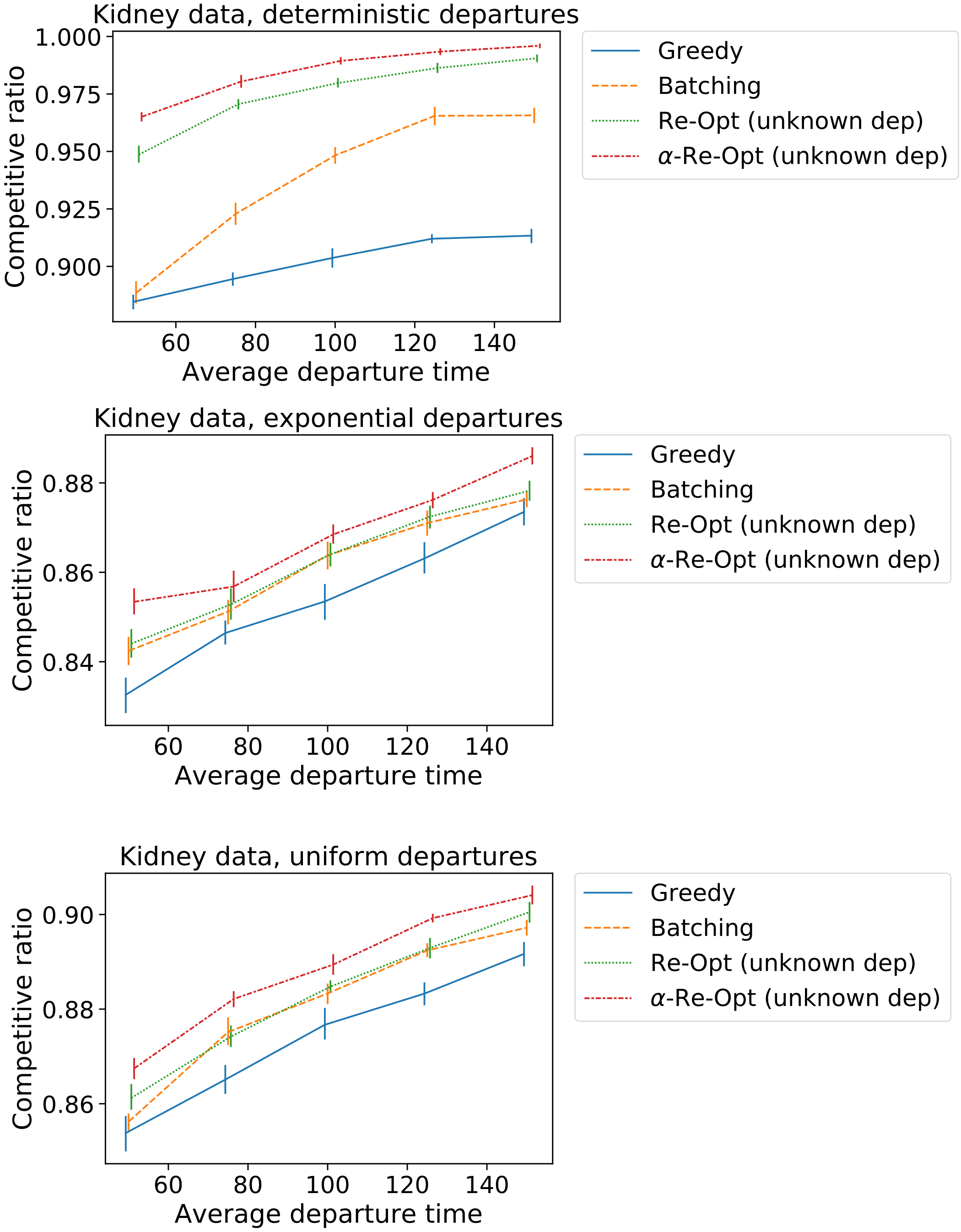

Figure 7-4: Performance of *Greedy*, *Batching*, *Re-Opt* and α-*Re-Opt* on kidney data. The compatibility graph G is binary-weighted.

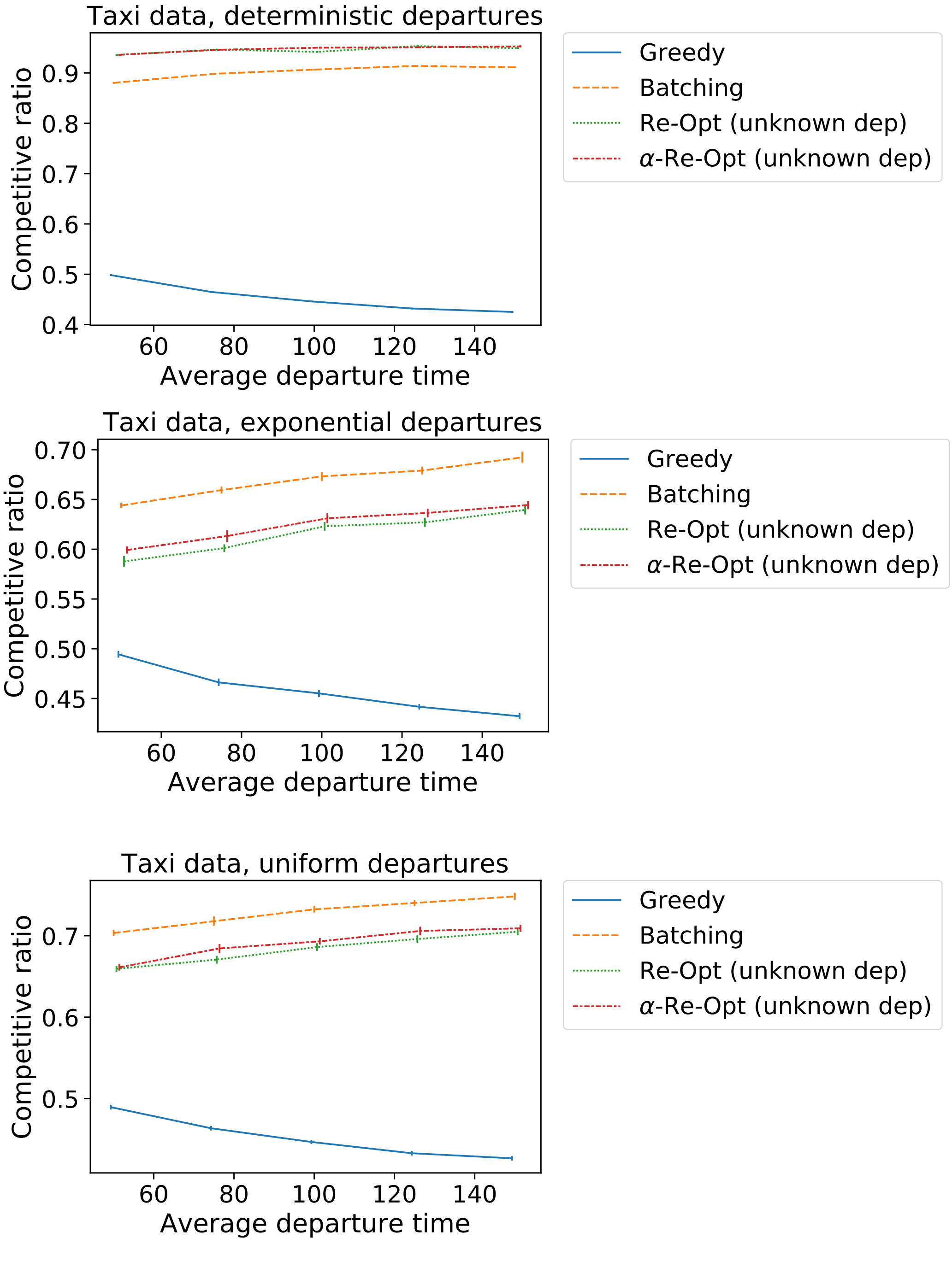

Figure 7-5: Performance of *Greedy*, *Batching*, *Re-Opt* and α-*Re-Opt* on taxi data. The compatibility graph G is weighted $\in \mathbb{R}$.

In Figure 7-5, we see that in the case of deterministic departures, the critical times are de facto known, and results are similar to those of section 7.4.2. In the case of either *exponential* or *uniform* departures, *Batching* seem to be performing significantly better than *Re-Opt*.

Next Steps

Two interesting areas for future work include the setting when the information about agent's departure times is uncertain, as well as models that are less restrictive than the adversarial setting (see, e.g., Ozkan and Ward [95]).

7.5 Conjecture: Batching is asymptotically optimal.

In this section we show how to reduce the analysis of batching to a graph-theoretic problem. We then provide numerical evidence for this problem, and evidence for the following conjecture:

Conjecture 2. *For n large enough,* Batching *is $1/2$-competitive when arrivals are in random order.*

Observe that it is not difficult to extend the example in Proposition 14 to a graph with arbitrary many vertices. In turn this proves our claim that asymptotically, as $n \to \infty$, the *Batching* algorithm has an optimal competitive ratio.

Proof Sketch and Key difficulty. Recall the notation from Section 7.2: $P_n^d(\sigma)$ denotes the offline graph under arrival permutation σ. Similarly, $B_n^d(\sigma)$ denotes the Batched graph under arrival permutation σ. *Offline* collects $\mathbb{E}\left[m(P_n^d(\sigma) * G)\right]$ while *Batching* collects $\mathbb{E}\left[m(B_n^d(\sigma) * G)\right]$.

Consider an edge $i, j \in [1, n]$. Let $p_{i,j} = \mathbb{P}_\sigma\left[(i,j) \in B_n^d | (i,j) \in P_n^d(\sigma)\right]$.

Claim 8. *For all i, j, $p_{i,j} = \frac{1}{2}$.*

Proof. Observe that $(i,j) \in P_n^d(\sigma) \Leftrightarrow |\sigma(i) - \sigma(j)| \leq d$. Note that conditional on $|\sigma(i) - \sigma(j)| = k$, we have $\mathbb{P}_\sigma\left[(i,j) \in B_n^d(\sigma)\right] = \frac{d+1-k}{d+1}$.

Using the fact that for all $k \in [1, d]$, $\mathbb{P}\left[|\sigma(i) - \sigma(j)| = k\right] = \frac{1}{d}$, this implies

$$p_{i,j} = \sum_{k=1}^{d} \mathbb{P}\left[|\sigma(i) - \sigma(j)| = k\right] \frac{d + 1 - k}{d + 1} = \frac{1}{2}.$$

□

Note however that because matching m is not a linear operator, we cannot conclude that $\mathbb{E}\left[m(B_n^d * G)\right] = \frac{1}{2}\mathbb{E}\left[(\right] m(P_n^d(\sigma) * G))$. Another way to see it is as follows: for a given σ, edges of different lengths have different probabilities of being kept in the *Batched* graph. The maximum-weight matching on $P_n^d(\sigma) * G$ may contain a higher fraction of long edges. □

In the next Section we try to formulate this key difficulty as a graph theoretic problem.

7.5.1 A graph-theoretic problem.

Consider the n-cycle to the power d: C_n^d (see e.g. Figure 7-6). Let $\mathcal{D}$ be a probability distribution with support over the non-negative real numbers. We sample an edge weight, i.i.d from $\mathcal{D}$, for each edge in C_n^d.

Now consider a maximum-weight matching over the resulting weighted graph. For every edge $(i, j) \in C_n^d$, let us denote $p_{i,j}$ the probability that the edge is in the maximum-weight matching. We will say that edge (i, j) has *length* $|i - j|$. Observe that by symmetry, any two edges of the same length have the same probability of being in the matching.

Conjecture 3. *For n large enough, every edge in the graph has the same probability of being selected.*

Note that the length of an edge is defined using a canonical numbering of the vertices of the graph C_n^d. However, Conjecture 3 is a property of the graph itself. The key difficulty is that different pairs of vertices will induce different distributions over their neighbors.

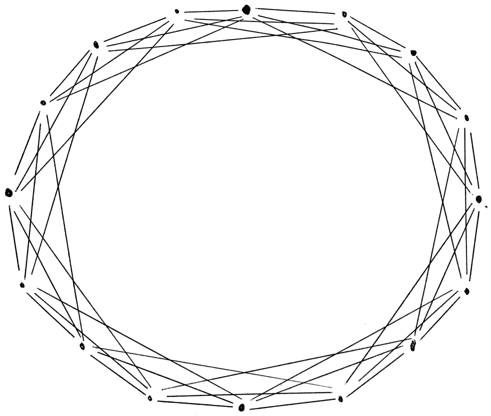

Figure 7-6: Representation of the n-Cycle to the power d, for $n = 16$ and $d = 3$.

7.5.2 Numerical Results

In Figure 7-7, we show that for a variety of distributions $\mathcal{D}$, the distribution of the lengths of edges selected in the maximum-weight matching appears to be uniform whenever n is large enough compared to d.

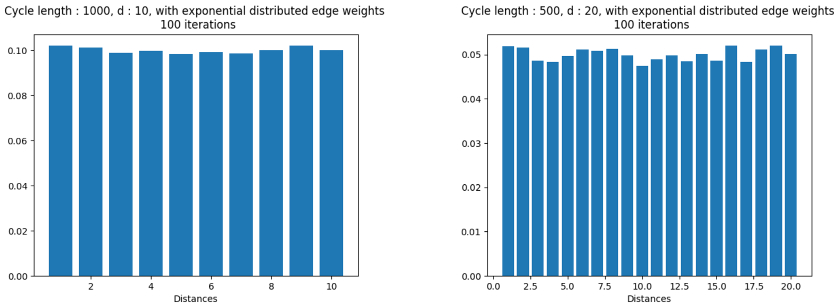

Figure 7-7: Distribution of the length of edges selected by the maximum-weight matching, for an exponential edge-weight distribution. Left: $n = 1000$ and $d = 10$. Right: $n = 500$ and $d = 20$. Both figures were conducted across 100 independent random draws and averaged.

We then ran a χ^2 statistical test to determine the probability that this distribution is uniform. We obtained p-values above 0.999 for $n \in [50, 100]$ and $d \in [5, 50]$. Although that is not a proof that the distribution is uniform, this reinforces our conjecture that the distribution of matched edge lengths is uniform.

7.6 Conclusion

In this chapter, we build on the model introduced in Chapter 6 by assuming that the vertices will not arrive in the worst possible order sequence. As a consequence, deterministic policies such as *Batching* are no longer garanteed to perform badly. Our main result is to show that under this new *random order of arrivals* assumption, the batching algorithm is 0.279-competitive. Our proof first reduces the problem to a special graph covering problem. We then show that an lower bound on the competitive ratio of *Batching* can be computed as the limit of a sequence of Linear Programs. Finally, we prove that it is sufficient to explicitely solve a small number of these LPs, and we provide a computer-aided argument to conclude.

We also extend our proof framework to the setting where the online algorithm is allowed to know which vertices are going to arrive l steps ahead of time. We provide a closed-form competitive ratio that depends on the ratio of l/d. In particular we show that if $l \geq d$, then the batching algorithm performs better than *any* online algorithm that does not have access to the look-ahead information.

Finally, we conjecture that *batching* is in fact 1/2-competitive, and we provide a proof sketch as well as some numerical insights and intuition. Finally, we provide numerical simulation results based on datasets of New York City's taxis and the National Kidney Registry. Our simulations compare *Batching* to the *Re-Optimization* algorithm introduced in Chapter 6.

Chapter 8

A Dynamic Matching Framework with Stochastic Information

8.1 Introduction

In this chapter, we study instances where the platform has access to additional information regarding future arrivals of vertices. This information can consist of a stochastic model, or historical data. In section 8.2 we model this historical data as a probabilistic model on the compatibility graph G and study algorithms that achieve a constant fraction of the optimal matching. In section 8.3 we show how the optimal algorithm can be formulated through dynamic programming.

8.1.1 Model

Consistent with the notation introduced in Chapter 1, we are concerned with matching vertices of a *compatibility graph* G. The specificity in this chapter, is that G is going to be generated stochastically, the matching algorithm is allowed additional information regarding the distribution from which G is sampled.

We define an edge-weighted *type* graph T. For each vertex x in T , we define an arrival rate λ_x and a departure rate δ_x. The algorithm knows T, $\boldsymbol{\lambda}$ and $\boldsymbol{\theta}$. We then generate the compatibility graph G as follows: for every x, vertices i_x arrive according to a Poisson(λ_x) process. We say that i_x is *of type* x and its departure time is sampled from an Exponential(θ_x) distribution. Upon arrival of a vertex i_x of type x, it has an edge towards every vertex j of type y if y is a neighbor of x in T.

For a given matching rule R, the triplet $(T, \boldsymbol{\lambda}, \boldsymbol{\theta})$ defines a distribution over instances of dynamic matching problems. Note that because our departure process is memoryless, at each time t, the graph of vertices present can be represented compactly through a vector $N(t) = (N_x(t))_{x \in T}$ that represents how many vertices of each type are present.

8.2 Competitive analysis

8.2.1 Analysis of the *Offline* algorithm $\mathcal{O}$.

Let us denote $v_{i,j}$ the weight of edge (i,j) in T, and let f be an optimal solution to the following "fractionnal assignment" linear program:

$$\begin{aligned}
\max \sum_{i<j} v_{i,j} f_{i,j} \\
\text{s.t.} \sum_{j} f_{i,j} \leq p_i \ \forall i \in T \\
f_{i,j} = f_{j,i} \in [0,1]
\end{aligned} \tag{8.1}$$

Proposition 16. $\mathcal{O} \leq \sum_{i<j} f_{i,j} v_{i,j}$.

Proof. Consider a realization of arrivals $a = (a_1, ..., a_N) \in T^N$. Let n_i be the number of arrivals of type t_i. If there are no departures, i.e. $\delta_i = 0$ for all i, then the offline solution can be found by solving the following problem:

$$\begin{aligned}
\max \sum_{i<j} v_{i,j} x_{i,j} \\
\text{s.t.} \sum_{j} x_{i,j} \leq n_i \ \forall i \in T \\
x_{i,j} = x_{j,i} \in \{0,1\}
\end{aligned} \tag{8.2}$$

Let $x^*(a)$ be the solution of the LP relaxation of 8.2. And let $\mathbf{g} = \sum_a x^*(a) \mathbb{P}[a]$ be the expected vector of matches. Note that when departures are not 0, the expected performance of the offline algorithm is given by:

$$\mathcal{O} \leq \frac{1}{N} \sum_{i,j} g_{i,j} w_{i,j}.$$

Note that $\sum_j g_{i,j} = \sum_a \left(\sum_j x^*(a)_{i,j}\right) \mathbb{P}[a] \leq \sum_a n_i \mathbb{P}[a] = \mathbb{E}[n_i] = Np_i$. Therefore g/N is a feasible solution to 8.1 which concludes the proof.

□

8.2.2 Special case: bipartite graphs with one-sided departures.

Consider the case where the type graph T is bipartite: vertices i belong to two sets A and B. Suppose that for all $i \in A$, $\delta_i = 0$. Let f be an optimal solution to the "fractional assignment" defined in (8.1).

Algorithm 6 Suggested Matching ($\mathcal{S}_\epsilon$)

For each arrival x of type $i \in B$: Sample a type $j \in A$ with probability $\frac{(1-\epsilon)f_{i,j}}{p_i}$. If $N_j > 0$, match x to a node of type j. Otherwise, discard node x.

Proposition 17. *$\mathcal{S}_\epsilon$ achieves a competitive ratio of* $(1-\epsilon)^2$.

Proof. Fix $j \in A$. Let I_j^t be the indicator that a node of type j arrives at t. Let $M_{i,j}^t$ be the indicator that $i \in B$ arrives at t and tries to match to j. Note that $\mathbb{P}\left[M_{i,j}^t = 1\right] = (1-\epsilon)f_{i,j}$. We have

$$N_i^{t+1} = N_i^t + I_i^t - \sum_{j \in B} M_{i,j}^t \mathbb{I}_{N_i^t > 0} \geq N_i^t + I_i^t - \sum_{j \in B} M_{i,j}^t.$$

Note that $X^t = I_i^t - \sum_{j \in B} M_{i,j}^t$ are i.i.d. random variables with mean $\mathbb{E}[X^t] = \epsilon p_i$, and are bounded between -1 and 1 (assuming exactly one arrival per time step). Therefore, for all t, $N_i^t \geq \sum_{k=0}^t X(k)$, and using Hoeffding's inequality, we get:

$$\mathbb{P}\left[N_i^t = 0\right] \leq \mathbb{P}\left[\sum_{k=0}^{t-1} X(k) \leq 0\right] \leq e^{-\epsilon^2 p_i^2 t/2}$$

Take t^* such that for all $i \in A$, $e^{-\epsilon^2 p_i^2 t^*/2} \leq \epsilon$. For all $t \geq t^*$, the probability that $j \in B$ matches $i \in A$ at t is given by $\mathbb{P}\left[M_{i,j}^t\right] \mathbb{P}\left[N_i^t > 0\right] \geq (1-\epsilon)^2 f_{i,j}$. Therefore, $\mathbb{E}[\mathcal{S}_\epsilon(t)] \geq (t - t^*)(1-\epsilon)^2 \sum_{i,j} f_{i,j} v_{i,j}$. □

8.2.3 An $\frac{1}{4(1+\gamma)}$-approximation algorithm

Let f be an optimal solution to (8.1).

Algorithm 7 Two-sided Suggested Matching ($\mathcal{S}_\alpha$)

For each arrival x of type i:

- With probability α, make x wait.
- With probability $1-\alpha$, try to match x: sample a type j with probability $\frac{f_{i,j}}{p_i}$. If there is a node of type t_j, match it to the arriving node. Otherwise, discard node x (i.e. never match it).

Theorem 8. *Let $\gamma = \max_i \frac{\delta_i}{p_i}$. Then, $\mathcal{S}_\alpha$ is $\frac{\alpha(1-\alpha)}{(1+\gamma)}$-competitive.*

Proof. Let N_i^t be the number of nodes of type i after k arrivals. From Lemma 5, we know that the Markov Chains N_i reach steady-state. Let τ be the maximum mixing time over all i. Using the notations from 17, let $M_{i,j}^t$ be the event that i tries to match to j at t. $\mathbb{P}[M_{i,j}^t = 1] = (1-\alpha)p_i\frac{f_{i,j}}{p_i}$, and Let $X_{i,j}^t = M_{i,j}^t \mathbb{I}_{N_j^t > 0}$ be the event that this try succeeds. We have $\mathbb{E}[\mathcal{S}_\alpha(t)] \geq \sum_{k=\tau+1}^{t} \sum_{i<j} \mathbb{P}\left[X_{i,j}\right] w_{i,j}$.

From Lemma 5 we know that $\mathbb{P}[X_{i,j}(t)] = (1-\alpha)f_{i,j}\mathbb{P}[N_i(t) > 0] = (1-\alpha)f_{i,j}\frac{\alpha p_i}{p_i+\delta_i}$. Therefore

$$\begin{aligned} \mathbb{E}[\mathcal{S}_\alpha(t)] &\geq (t-\tau)\sum_{i,j}(1-\alpha)f_{i,j}\frac{\alpha p_i}{p_i+\delta_i}w_{i.j} \\ &\geq \frac{\alpha(1-\alpha)(t-\tau)}{(1+\gamma)}\sum_{i,j} f_{i,j}w_{i,j} \end{aligned} \tag{8.3}$$

Which concludes the proof. □

Lemma 5. *For all i, the process $(N_i^t)_{t\geq 0}$ is a positive recurrent Markov chain which reaches steady-state distribution $\pi_i = (\pi_i^0, \pi_i^1, ...)$, and in steady-state, $\mathbb{P}[N_i(t) > 0] \geq \frac{\alpha p_i}{p_i+\delta_i}$.*

Proof. Let us fix a type i. At each time step t, let A_t be the indicator that a node of type i arrives and has to wait, thus $\mathbb{P}[A_i^t = 1] = \alpha p_i$. Let B_i^t be the indicator that a node of type i

gets matched by the incoming node at t: the probability that a node of type j tries to match a node of type i is $(1-\alpha)p_j \frac{f_{i,j}}{p_j}$, which means that $\mathbb{P}\left[B_i^t = 1\right] = (1-\alpha)\mathbb{I}_{N_i^t > 0} \sum_j p_j \frac{f_{i,j}}{p_j} = \mathbb{I}_{N_i^t > 0}(1-\alpha)p_i$. Finally, let D_i^t be the number of departures at k, $D_t \sim Bin(N_i^t + A_i^t - B_i^t, \delta_i)$. Note that we have:

$$N_i^{t+1} = N_i^t + A_i^t - B_i^t - D_i^t$$

which proves that N^i has the Markov property. Furthermore if $N_i^t \geq 2$ then $N_i^t + A_i^t - B_i^t \geq 1$ and $\mathbb{E}[D_i^t] \geq \delta_i$. Assuming $\alpha \leq 1/2$, $\mathbb{E}[N_i^{t+1} - N_i^t] = \alpha p_i - (1-\alpha)p_i - \delta_i \leq -\delta_i$. Using a result by Foster, this proves that N_i is positive recurrent.

In steady-state, flow conservation yields:

$$\begin{aligned}
\pi_i^0 \alpha p_i &= \sum_{k \geq 1} \pi_i^k \mathbb{P}[N_i^{t+1} = 0 | N_i^t = k] \\
&= \pi_i^1((1-\alpha)p_i + (1-(1-\alpha)p_i)\delta_i) + \sum_{k \geq 2} \pi_i^k \mathbb{P}[N_i^{t+1} = 0 | N_i^t = k] \\
&\leq \pi_i^1((1-\alpha)p_i + \delta_i) + \sum_{k \geq 2} \pi_i^k \delta_i \\
&\leq \sum_{k \geq 1} \pi_i^k((1-\alpha)p_i + \delta_i) = (1-\pi_i^0)((1-\alpha)p_i + \delta_i)
\end{aligned} \tag{8.4}$$

Therefore $\pi_i^0 \leq \frac{(1-\alpha)p_i + \delta_i}{p_i + \delta_i}$. □

8.3 Dynamic programming and Reinforcement Learning

In this section, we change focus from trying to find algorithms which perform well when compared to the *Offline* algorithm $\mathcal{O}$, and instead we try to approximate the optimal online algorithm.

Efficient priority computation

In Chapter 3 we described three priority mechanisms that resemble those used in practice by kidney exchange programs. By and large, these priority mechanisms are designed to achieve two goals. The first is to increase the odds of socially desirable outcomes (such as a child receiving a kidney). This is a consideration that is independent from the dynamic nature of the problem, and we model this through the weights in the compatibility graph (more desirable outcomes receive a higher weight).

The second objective is to prevent suboptimal matches that would occur under a greedy algorithm.[1] While these priorities are typically computed manually, there have also been attempts to compute them through exhaustive search methods [45]. In chapter 7, we introduce a new idea of *opportunistic batching* which tries to keep some vertices waiting when they have a directed edge towards a hard-to-match agent. This "dynamic" priority system differs from a "static" priority system where waiting and priorities are determined by an agent's characteristics rather than by the compatibility graph configuration.

Our main idea in this section is to formulate a flexible yet powerful formulation that enables us to automatically compute the priorities in order to prevent sub-optimal matches.

Markov decision process

Recall that when the departure process is memoryless, the state of the system can be compactly represented by the vector $\mathbf{N}(t) = (N_x(t))_{x \in T}$, where $N_x(t)$ represents the number of vertices of type x at time t. In Dynamic Programming terms, this means that our dynamic matching problem is a (fully observable) Markov Decision Process. We will say that $\boldsymbol{N}$ is the the *state* of the system and belongs to the *state space* $\mathcal{S} = \mathbb{N}^{|T|}$.

For a given matching rule R, we will say that $A \in \mathcal{S}$ is a feasible allocation for state $\boldsymbol{N}$ if and only if for all x, $A_x \leq N_x$, and there exists a set $m_1, ..., m_k$ of disjoint valid matchings (according to rule R) such that $\sum_k \mathbb{I}_{x \in m_k} = A_x$. We will say that the set of all such feasible allocations $\mathcal{A}(\boldsymbol{N})$ is the decision space at state $\boldsymbol{N}$.

Observe that in this setting, we can relax the assumption that the graph needs to be undirected. We can also consider more flexible matching rules, and allow for instance several participants to trade together.

[1] For example, there is concern that blood type O donor may inefficiently donate to a blood type A patient

8.3.1 Approximating the optimal value function

Recall that we use $v(A)$ to denote the value of an allocation M. Using the notation above, the greedy algorithm solves at each time step the following maximization problem:

$$\max_{A \in \mathcal{A}} v(A). \tag{8.5}$$

Optimal Online Algorithm

For any algorithm, we can define an associated value function $J : \mathcal{S} \mapsto \mathbb{R}$. For any state $\boldsymbol{N}$, $J(\boldsymbol{N})$ represents the expected value from using the algorithm starting in state $\boldsymbol{N}$. Although we are interested in the algorithm that maximizes the expected reward on average in the long run, results from the Dynamic Programming literature [124] allow us to consider the discounted version of the problem with a discount factor δ close to 1.

We can further define the *optimal* value function implicitely using Bellman's equation:

$$\begin{aligned} J^*(N(t)) &= \max_{A \in \mathcal{A}(\boldsymbol{N}(t))} v(A) + \delta \mathbb{E}\left[J^*(N(t+1))\right] \\ &= \max_{A \in \mathcal{A}(\boldsymbol{N}(t))} v(A) + \delta \mathbb{E}\left[J^*\left(N(t) - D(N(t) - A) - A\right)\right] \end{aligned} \tag{8.6}$$

Where $D(N(t) - A)$ is a random variable corresponding to the arrivals between times t and $t+1$, minus the departures of unmatched vertices. We use the convention that if $S \notin \mathbf{N}^I$ then $J(S) = -\infty$.

The key difficulty in finding an approximate solution to (8.6) is twofold: the state $\mathcal{S}$ can be very high dimensional (when there are many different types in T), and the decision space $\mathcal{A}$ is even larger and has a combinatorial structure. In the rest of this section we will explore different ideas from the approximate dynamic programming and the reinforcement learning literature, and see how they perform empirically.

Linear Value Function Approximation

One simple idea is to approximate J^* with a function $\widetilde{J}(S)$ that is linear in $\boldsymbol{N}$. We then compute for each type x a *shadow price* v_x, and set $\widetilde{J}(N) = \sum_x v_x N_x$. We can think of v_x as the discounted expected value of having a vertex of type x in the graph in the future. Assuming the prices $\boldsymbol{v}$ are known, this leads to a nice simplification of (8.6):

$$\widetilde{J}(N(t)) = \max_{A \in \mathcal{A}(N(t))} v(A) + \delta \mathbb{E}\left[\widetilde{J}(N(t) - D(N(t) - A) - A)\right]. \tag{8.7}$$

$$= \max_{A \in \mathcal{A}(N(t))} \left(v(A) - \sum_x \delta(1-\theta_x) v_x A_x \right) + \delta \mathbb{E}\left[\widetilde{J}(N(t) - D(N(t)))\right]. \tag{8.8}$$

Where we used the fact that $\mathbb{E}\left[D(N-A)\right] = \mathbb{E}\left[D(N)\right] - \boldsymbol{\theta} A$, the linearity of $\widetilde{J}$, and the linearity of expectation. This means that conditional on knowing the values v_x, each allocation can be found by solving the following one stage problem:

$$A = \text{argmax}_{A \in \mathcal{A}(N(t))} v(A) - \delta \sum_x (1-\theta_x) A_x v_x \tag{8.9}$$

If the original "greedy" problem in (8.5) can be framed as a linear (or mixed integer) program, then so can the problem in (8.9). See for example (8.2).

Computing the shadow prices

Computing $\boldsymbol{v}$ can be done using the temporal difference (TD) algorithm. We can measure the error based on Bellman's equation:

$$\epsilon_t = c^T \mathbf{x}_t + \delta J(N_{t+1}) - J(N_t)$$

Where $N_{t+1} = N_t - A_t - D(N_t - A_t)$. The next step is to update the shadow prices according to a stochastic gradient descent step:

$$\begin{aligned} v_{t+1}^i &= v_t^i + \alpha_t \epsilon_t \nabla_{\mathbf{v}} J(S_t) \\ &= v_t^i + \alpha_t \epsilon_t N_i(t). \end{aligned} \tag{8.10}$$

Where α_t is a step-size satisfying the Robbins-Monro conditions [2]. Note that this method does not require knowledge of the arrival and departure process to function.

8.3.2 Extension: learning from covariates

One difficulty of the previous approach is the necessity of learning a value v^i for each type. Deciding on how to define the different types can be quite difficult and this approach does not leverage the idea that some types may share certain characteristics. To tackle this problem, we extend our model in the following way.

At each time step $t = 0, ..., N$, a set of arriving nodes A_t is sampled from a distribution $\mathcal{D}$. Each node $n \in A_t$ has a set of characteristics $\psi(n) \in \mathbb{R}^k$. Matching nodes n_1 to n_2 (at any time) has value $c(\psi(n_1), \psi(n_2)) \in \mathbb{R}$. At the end of each time period, any unmatched node n departs with probability $\delta(\psi(n))$.

Value function estimator

Again we can assume an additive value function $\widetilde{J}(S) = \sum_{n \in S} v_\theta(\psi(n))$. We can think of v_θ as an estimator of the value of a node with characteristics $\psi(n)$. In this case, v depends on a set of parameters θ to be optimised.

Because we assume the same estimator for all nodes, this allows us to speed-up the learning process by taking advantage of possible similarities between incoming nodes. (8.10)

[2] $\sum \alpha_t = \infty, \sum \alpha_t^2 < \infty$

becomes:

$$\begin{aligned}\theta_{t+1} &= \theta_t + \alpha_t \epsilon_t \nabla_{\mathbf{v}} J(S_t) \\ &= \theta_t + \alpha_t \epsilon_t \sum_{n \in S_t} \frac{\partial v}{\partial \theta}(\psi(n)).\end{aligned} \tag{8.11}$$

8.4 Conclusion

8.4.1 Key contributions

In this Chapter we introduce a model for dynamic matching with stochastic information. We provide analysis in terms of competitive ratio for a simple *Suggested Matching* algorithm that uses offline statistics to guide the online matching.

In a special case where the compatibility graph is bipartite, and vertices on one side never depart, we show that *SM* is optimal. This setting generalizes the traditional *online matching* problem in the stochastic case, with applications to online advertising.

In the case of a general graph, we are not able to prove a constant competitive ratio. We provide a competitive ratio that depends on the largest arrival-to-departure ratio among possible types of vertices. We expect that a constant-competitive algorithm would need to use departure information.

We next study this problem using the lens of Dynamic Programming. This then allows us to extend our model to the case where participant characteristics can take continuous values (rather than a simple grouping of participants into discrete *types*).

8.4.2 Future directions

Dynamic matching problems with stochastic information are at the interface between the fields of online optimization and online learning.

Theory

From a theoretical perspective, finding a constant-competitive algorithm would show that having access to stochastic information can offset the issue of heterogeneous departure rates.

In the special case where all types have the same departure rate, finding a 0.5-competitive algorithm would be a significant step with respect to the *Random Order* setting. In particular, this would highlight the value of knowing the underlying arrival distribution.

Practice

In terms of improving dynamic matching algorithms in practice, a few interesting future directions include: utilizing participant covariates through better learning algorithms. This would help bridge the theory-practice gap and enable useful algorithms for practitioners. Adding richer matching constraints, such as long cycles, hypergraph matching, etc, would also be useful to help bring these algorithm in practical settings.

Finally, having a common set of benchmarks would enable researchers to compare their results and introduce new algorithms for which theoretical bounds are hard to prove.

Chapter 9

Concluding Remarks

To conclude this dissertation we provide a summary of our work, in terms of its practical implications as well as our technical contributions. We also provide a few directions for future research.

9.1 Contributions of this Dissertation

9.1.1 Technical contributions

In Chapter 4 we introduce a theoretical model that accounts for both patient-donor heterogeneity and the difficulty that some patients have in finding a suitable donor before they need to depart. Importantly, we make sure that our model still remains tractable analytically.

In Chapter 5 we study a random-graph model of dynamic matching with no departures. While all agents are potentially compatible with each other, some are hard-to-match and others are easy-to-match. We use a new method for the analysis of countable-state, two-dimensional Markov chains. We adopt an asymptotic approach and use this method to prove tight asymptotic bounds on waiting times, under myopic policies that differ in matching technology and prioritization.

In Chapter 6 we introduce a new theoretical model for edge-weighted matching when all vertices arrive online and leave after d time periods. Our main contribution is a 0.25-competitive algorithm. The algorithm randomly selects a subset of agents who will wait until right before their departure to get matched, and maintains a maximum-weight matching with respect to the other agents. We show this through a reduction to a *bipartite constrained* online matching problem, which may be of independent interest. The primal-dual analysis of the algorithm hinges on a careful comparison between the initial dual value associated with an agent when it first arrives, and the final value after d time steps. We also show that no algorithm is more than 0.5-competitive. We extend the model to the case in which departure times are drawn independently from an exponential distribution, and extend our analysis to establish a 0.125-competitive algorithm in this setting.

In Chapter 7 we study a similar setting where vertices arrive in a random order and leave after d time periods. Our main result is to show that the *batching* algorithm, which periodically searches for the highest-value allocation, is 0.279-competitive. Our proof first reduces the problem to a special graph covering problem. We then show that a lower bound on the competitive ratio of *Batching* can be computed as the limit of a sequence of Linear Programs. Finally, we prove that it is sufficient to explicitly solve a small number of these Linear Programs, and we provide a computer-aided argument to conclude. We also extend our proof framework to the setting where the online algorithm is allowed to know which vertices are going to arrive l steps ahead of time. We provide a closed-form competitive ratio that depends on the ratio of $^l/_d$. In particular we show that if $l \geq d$, then the batching algorithm performs better than *any* online algorithm that does not have access to the look-ahead information.

In Chapter 8 we consider the setting where the compatibility graph is sampled from a distribution which is known to the matching algorithm. We formalize this using a *type* graph, from which incoming vertices are sampled. We first provide special cases and corresponding constant-competitive algorithms. We then frame the problem through the lens of dynamic programming. We finally show how ideas from approximate dynamic programming can be leveraged to provide useful algorithms.

9.1.2 Practical insights

In Chapter 3 we conducted simulations using clinical data from two Kidney Exchange programs in the US to study how the frequency of match-runs impacts the number of transplants and the average waiting times. We simulate the options facing each of the two programs by repeated resampling from their historical pools of patient-donor pairs and non-directed donors, with arrival and departure rates corresponding to the historical data. We find that longer intervals between match-runs do not increase the total number of transplants, and that prioritizing highly sensitized patients is more effective than waiting longer between match-runs for transplanting highly sensitized patients. While we do not find that frequent match-runs result in fewer transplanted pairs, we do find that increasing arrival rates of new pairs improves both the fraction of transplanted pairs and waiting times.

In Chapter 4 we study how both the priority structure, and the length of cycles impact the choice of the optimal match frequency. We find that the benefits of small batches have negligible effect on the number of pairs who can give to hard-to-match patients. In other words, their in-degree in the compatibility graph stays small. Subsequently, when only cycles of length 2 are allowed, waiting does not result in a significant number of additional matches. Surprisingly, we find a different result when cycles of length up to 3 are allowed. Although the degree of hard-to-match pairs is still small, it is not always zero. In some cases, they are compatible to receive from another pair but are not able to give back, precluding a 2-cycle. In that case, waiting for a third pair to arrive may increase the odds of closing the loop. This suggests that in some circumstances, some amount of *targeted* waiting may be beneficial, even when making *everyone* wait is counter-productive.

In Chapter 5 we find that the market composition is a key factor in the desired matching technology and prioritization level. When hard-to-match agents arrive less frequently than easy-to-match ones (i) bilateral matching is almost as efficient as chains, and (ii) assigning priorities to hard-to-match agents improves their waiting times. When hard-to-match agents arrive more frequently, chains are much more efficient than bilateral matching and prioritization has no impact. We further study the effect of arrival rates on the average waiting time. Somewhat surprisingly, we find that in a heterogeneous market and under bilateral matching, increasing arrival rate of hard-to-match agents has a non-monotone effect on waiting times, due to the fact that, under some market compositions, there is an adverse effect of competition. Our comparative statics shed light on the impact of merging markets and attracting altruistic agents (that initiate chains) or easy-to-match agents. This work uncovers fundamental differences between heterogeneous and homogeneous dynamic markets, and potentially helps policy makers to generate insights on the operations of matching markets such as kidney exchange programs.

In Chapter 6 we introduce new algorithms for dynamic matching. The fact that the *DDA* algorithm performs well in the worst case setting rationalizes the *re-optimization*, or *rolling-horizon* technique often used in practice (See Section 2.4 in particular). Similarly, although we are not able to show improved competitive ratios for the *α-DDA* algorithm, the good empirical performance can be seen as a justification to give priorities to participants who have been waiting longer.

In Chapters 7 and 8, we provide numerical simulation results based on datasets of New York City's taxis and the National Kidney Registry. Our simulations compare *Batching* to the *Re-Opt* algorithm introduced in Chapter 6. We find that in settings where exact departure times are known, *Re-Opt* significantly outperforms both *Batching* and *Greedy*. When only the departure distribution is known, then *Re-Opt* outperforms *Batching* on kidney data, but not on taxi data. Finally, we show that borrowing ideas from *α-DDA* in Chapter 6, we can often improve results of *Re-Opt*.

9.2 Directions for future fesearch

Extensions to new settings

While our focus was on *Kidney Exchange* and *Carpooling*, some new models and results would be very interesting in other *centralized* settings. Beyond this, our theoretical models in Chapters 6 and 7 could also be extended to model *decentralized* platforms where participants are endowed with a probabilistic model for accepting one matches from a suggested set.

New analysis of existing algorithms

In Chapter 6, we conjecture that our *DDA* algorithm can be extended to achieve $\frac{\sqrt{5}-1}{2}$ competitive ratio on *bipartite constrained graphs*. We also think that the *PG, DDA* and *α-DDA* algorithms could also be interesting to study in the random arrival setting.

In Chapter 7, we conjecture that *Batching* is in fact 0.5-competitive. While the analysis is likely to be involved, proving this would provide a strong justification for practical use of this simple algorithm.

New algorithms and models

In Chapter 6, we study algorithms that reduce the problem to a *bipartite constrained* graph. New algorithms that do not require such a reduction could lead to improved performance guarantees.

We prove that if departures are determined adversarially no algorithm is constant-competitive. Nonetheless, an interesting line of research would be to design algorithm that perform well on stochastic models of departures. Similarly, it would be interesting to study stochastic departures in the random arrival order setting.

Bibliography

[1] I. Adan and G. Weiss. Exact FCFS Matching Rates for Two Infinite Multitype Sequences. *Operations Research*, 60(2):475–489, 2012.

[2] Nikhil Agarwal, Itai Ashlagi, Michael Rees, Paulo Somaini, and Daniel Waldinger. An empirical framework for sequential assignment: The allocation of deceased donor kidneys, 2018.

[3] Niels Agatz, Alan Erera, Martin Savelsbergh, and Xing Wang. Optimization for dynamic ride-sharing: A review. *European Journal of Operational Research*, 223(2): 295–303, 2012.

[4] M. Akbarpour, S. Li, and S. Oveis Gharan. Dynamic matching market design. Working paper, 2014.

[5] Mohammad Akbarpour, Shengwu Li, and Shayan Oveis Gharan. Thickness and information in dynamic matching markets. 2017.

[6] Javier Alonso-Mora, Samitha Samaranayake, Alex Wallar, Emilio Frazzoli, and Daniela Rus. On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *Proceedings of the National Academy of Sciences*, 114(3):462–467, 2017.

[7] R. Anderson, I. Ashlagi, D. Gamarnik, M. Rees, A.E. Roth, T. Sönmez, and M.U. Ünver. Kidney exchange and the alliance for paired donation: Operations research changes the way kidneys are transplanted. *Interfaces*, 45(1):26–42, 2015.

[8] R. Anderson, I. Ashlagi, Y. Kanoria, and D. Gamarnik. Efficient dynamic barter exchange. Forthcoming in Operations Research, 2017.

[9] Ross Anderson, Itai Ashlagi, David Gamarnik, and Yash Kanoria. A dynamic model of barter exchange. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*, pages 1925–1933. Society for Industrial and Applied Mathematics, 2015.

[10] D. Angluin and L.G. Valiant. Fast probabilistic algorithms for hamiltonian circuits and matchings. *Journal of Computer and System Sciences*, 18(2):155 – 193, 1979.

[11] Nick Arnosti and Peng Shi. How (not) to allocate affordable housing. 2017.

[12] I. Ashlagi and A. E. Roth. Free riding and participation in large scale, multi-hospital kidney exchange. *Theoretical Economics*, 9(3):817–863, 2014.

[13] I. Ashlagi and A.E. Roth. New Challenges in Multi-hospital Kidney Exchange. *American Economic Review, Papers and Proceedings*, 102(3):354–359, 2012.

[14] I. Ashlagi, D. Gamarnik, M. Rees, and A. Roth. The need for (long) chains in kidney exchange. Technical report, National Bureau of Economic Research, 2012.

[15] Itai Ashlagi, Patrick Jaillet, and Vahideh H Manshadi. Kidney exchange in dynamic sparse heterogenous pools. *arXiv preprint arXiv:1301.3509*, 2013.

[16] Itai Ashlagi, Yossi Azar, Moses Charikar, Ashish Chiplunkar, Ofir Geri, Haim Kaplan, Rahul Makhijani, Yuyi Wang, and Roger Wattenhofer. Min-cost bipartite perfect matching with delays. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 81. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

[17] Itai Ashlagi, Adam Bingaman, Maximilien Burq, Vahideh Manshadi, David Gamarnik, Cathi Murphey, Alvin E Roth, Marc L Melcher, and Michael A Rees. The effect of match-run frequencies on the number of transplants and waiting times in kidney exchange. *American Journal of Transplantation*, 2017.

[18] Itai Ashlagi, Maximilien Burq, Patrick Jaillet, and Vahideh Manshadi. On matching and thickness in heterogeneous dynamic markets. 2017.

[19] Itai Ashlagi, Maximilien Burq, Chinmoy Dutta, Patrick Jaillet, Amin Saberi, and Chris Sholley. Maximum weight online matching with deadlines. *arXiv preprint arXiv:1808.03526*, 2018.

[20] Itai Ashlagi, Maximilien Burq, Patrick Jaillet, and Amin Saberi. Maximizing efficiency in dynamic matching markets. *arXiv preprint arXiv:1803.01285*, 2018.

[21] P. Awasthi and T. Sandholm. Online stochastic optimization in the large: application to kidney exchange. In *Proceedings of the 21st international jont conference on Artifical intelligence*, IJCAI'09, pages 405–411, San Francisco, CA, USA, 2009. Morgan Kaufmann Publishers Inc. URL `http://dl.acm.org/citation.cfm?id=1661445.1661510`.

[22] Mariagiovanna Baccara, SangMok Lee, and Leeat Yariv. Optimal dynamic matching. 2018.

[23] Siddhartha Banerjee, Daniel Freund, and Thodoris Lykouris. Pricing and optimization in shared vehicle systems: An approximation framework. *arXiv preprint arXiv:1608.06819*, 2016.

[24] Siddhartha Banerjee, Yash Kanoria, and Pengyu Qian. State dependent control of closed queueing networks. In *Abstracts of the 2018 ACM International Conference on Measurement and Modeling of Computer Systems*, pages 2–4. ACM, 2018.

[25] Amotz Bar-Noy, Reuven Bar-Yehuda, Ari Freund, Joseph Naor, and Baruch Schieber. A unified approach to approximating resource allocation and scheduling. *Journal of the ACM (JACM)*, 48(5):1069–1090, 2001.

[26] Dimitri P Bertsekas. The auction algorithm: A distributed relaxation method for the assignment problem. *Annals of operations research*, 14(1):105–123, 1988.

[27] Dimitris Bertsimas, Patrick Jaillet, and Sébastien Martin. Online vehicle routing: The edge of optimization in large-scale applications. 2018.

[28] Adam W Bingaman, Francis H Wright, and Cathi L Murphey. Kidney paired donation in live-donor kidney transplantation. *New England Journal of Medicine*, 363(11):1091–1092, 2010.

[29] Benjamin Birnbaum and Claire Mathieu. On-line bipartite matching made simple. *ACM SIGACT News*, 39(1):80–87, 2008.

[30] F. Bloch and D. Cantala. Dynamic allocation of objects to queuing agents: The discrete model. 2014.

[31] A. Blum, A. Gupta, A. Procaccia, and A. Sharma. Harnessing the power of two crossmatches. In *Proceedings of the fourteenth ACM conference on Electronic commerce*, EC '13, pages 123–140, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1962-1. doi: 10.1145/2482540.2482569. URL `http://doi.acm.org/10.1145/2482540.2482569`.

[32] G. A. Böhmig, J. Fronek, A. Slavcev, G. F. Fischer, G. Berlakovich, and O. Viklicky. Czech-austrian kidney paired donation: first european cross-border living donor kidney exchange. *Transplant International*, 30(6):638–639, 2017.

[33] Ana Buŷić and Sean Meyn. Approximate optimality with bounded regret in dynamic matching models. *ACM SIGMETRICS Performance Evaluation Review*, 43(2):75–77, 2015.

[34] R. Caldentey, E. H. Kaplan, and G. Weiss. Fcfs infinite bipartite matching of servers and customers. *Adv. Appl. Prob*, 41(3):695–730, 2009.

[35] Juan Camilo Castillo, Dan Knoepfle, and Glen Weyl. Surge pricing solves the wild goose chase. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, pages 241–242. ACM, 2017.

[36] Francis YL Chin, Marek Chrobak, Stanley PY Fung, Wojciech Jawor, Jiří Sgall, and Tomáš Tichŷ. Online competitive algorithms for maximizing weighted throughput of unit jobs. *Journal of Discrete Algorithms*, 4(2):255–276, 2006.

[37] Jean-François Cordeau and Gilbert Laporte. The dial-a-ride problem: models and algorithms. *Annals of operations research*, 153(1):29–46, 2007.

[38] William G Couser, Giuseppe Remuzzi, Shanthi Mendis, and Marcello Tonelli. The contribution of chronic kidney disease to the global burden of major noncommunicable diseases. *Kidney international*, 80(12):1258–1270, 2011.

[39] Marry De Klerk, Karin M Keizer, Frans HJ Claas, Marian Witvliet, Bernadette JJM Haase-Kromwijk, and Willem Weimar. The dutch national living donor kidney exchange program. *American Journal of Transplantation*, 5(9):2302–2305, 2005.

[40] Marry de Klerk, Marian D Witvliet, BJ Haase-Kromwijk, Willem Weimar, and FH Claas. A flexible national living donor kidney exchange program taking advantage of a central histocompatibility laboratory: the dutch model. *Clin Transpl*, 69:73, 2008.

[41] Gabrielle Demange, David Gale, and Marilda Sotomayor. Multi-item auctions. *Journal of Political Economy*, 94(4):863–872, 1986.

[42] Nikhil R Devanur, Kamal Jain, and Robert D Kleinberg. Randomized primal-dual analysis of ranking for online bipartite matching. In *Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms*, pages 101–107. Society for Industrial and Applied Mathematics, 2013.

[43] J. P. Dickerson, A. D. Procaccia, and T. Sandholm. Optimizing Kidney Exchange with Transplant Chains: Theory and Reality. *Proc of the eleventh international conference on autonomous agents and multiagent systems*, 2012.

[44] J. P. Dickerson, A. D. Procaccia, and T. Sandholm. Dynamic Matching via Weighted Myopia with Application to Kidney Exchange. *Proc of the6th AAAI Conference on Artificial Intelligence*, pages 1340–1346, 2012.

[45] John P Dickerson, Ariel D Procaccia, and Tuomas Sandholm. Dynamic matching via weighted myopia with application to kidney exchange. In *AAAI*, 2012.

[46] John P Dickerson, Ariel D Procaccia, and Tuomas Sandholm. Failure-aware kidney exchange. In *Proceedings of the fourteenth ACM conference on Electronic commerce*, pages 323–340. ACM, 2013.

[47] John P Dickerson, Karthik A Sankararaman, Aravind Srinivasan, and Pan Xu. Allocation problems in ride-sharing platforms: Online matching with offline reusable resources. *arXiv preprint arXiv:1711.08345*, 2017.

[48] John P Dickerson, Ariel D Procaccia, and Tuomas Sandholm. Failure-aware kidney exchange. *Management Science*, 2018.

[49] Y. Ding, D. Ge, S. He, and C. T. Ryan. A non-asymptotic approach to analyzing kidney exchange graphs. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, pages 257–258. ACM, 2015.

[50] L. Doval. A theory of stability in dynamic matching markets. Technical report, mimeo, 2014.

[51] Jack Edmonds. Maximum matching and a polyhedron with 0, 1-vertices. *Journal of research of the National Bureau of Standards B*, 69(125-130):55–56, 1965.

[52] Paul W Eggers. Has the incidence of end-stage renal disease in the USA and other countries stabilized? *Current opinion in nephrology and hypertension*, 20(3):241–245, 2011.

[53] Yuval Emek, Shay Kutten, and Roger Wattenhofer. Online matching: haste makes waste! In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 333–344. ACM, 2016.

[54] J. Feldman, A. Mehta, V. S. Mirrokni, and S. Muthukrishnan. Online stochastic matching: Beating 1-1/e. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 117–126, 2009.

[55] Jon Feldman, Nitish Korula, Vahab Mirrokni, S Muthukrishnan, and Martin Pál. Online ad assignment with free disposal. In *International Workshop on Internet and Network Economics*, pages 374–385. Springer, 2009.

[56] Paolo Ferrari, Claudia Woodroffe, and Frank T Christiansen. Paired kidney donations to expand the living donor pool: the western australian experience. *Medical Journal of Australia*, 190(12):700, 2009.

[57] Paolo Ferrari, Willem Weimar, Rachel J Johnson, Wai H Lim, and Kathryn J Tinckam. Kidney paired donation: principles, protocols and programs. *Nephrology Dialysis Transplantation*, 30(8):1276–1285, 2014.

[58] Daniel Fershtman and Alessandro Pavan. Re-matching, experimentation, and cross-subsidization. Technical report, mimeo, 2015.

[59] DE Fumo, V Kapoor, LJ Reece, SM Stepkowski, JE Kopke, SE Rees, C Smith, AE Roth, AB Leichtman, and MA Rees. Historical matching strategies in kidney paired donation: The 7-year evolution of a web-based virtual matching system. *American Journal of Transplantation*, 15(10):2646–2654, 2015.

[60] D. Gale and L. L. Shapley. College Admissions and the Stability of Marriage. *American Mathematical Monthly*, 69:9–15, 1962.

[61] G. Goel and A. Mehta. Online budgeted matching in random input models with applications to adwords. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms (SODA)*, pages 982–991, 2008.

[62] Itai Gurvich and Amy Ward. On the dynamic control of matching queues. *Stochastic Systems*, 4(2):479–523, 2014.

[63] Bruce Hajek. On the competitiveness of on-line scheduling of unit-length packets with hard deadlines in slotted time. In *Proceedings of the 2001 Conference on Information Sciences and Systems*, 2001.

[64] Ruthanne L Hanto, William Reitsma, and Francis L Delmonico. The development of a successful multiregional kidney paired donation program. *Transplantation*, 86(12): 1744–1748, 2008.

[65] Ming Hu and Yun Zhou. Dynamic type matching. 2016.

[66] Zhiyi Huang, Ning Kang, Zhihao Gavin Tang, Xiaowei Wu, Yuhao Zhang, and Xue Zhu. How to match when all vertices arrive online. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 17–29. ACM, 2018.

[67] P. Jaillet and X. Lu. Online stochastic matching: New algorithms with better bounds. *Mathematics of Operations Research*, 39(3):624–646, 2013.

[68] Łukasz Jeż, Fei Li, Jay Sethuraman, and Clifford Stein. Online scheduling of packets with agreeable deadlines. *ACM Transactions on Algorithms (TALG)*, 9(1):5, 2012.

[69] Rachel J Johnson, Joanne E Allen, Susan V Fuggle, J Andrew Bradley, Chris Rudge, et al. Early experience of paired living kidney donation in the united kingdom. *Transplantation*, 86(12):1672–1677, 2008.

[70] S.V. Kadam and M.H. Kotowski. Multi-period matching. Technical report, Mimeo, 2014.

[71] Edward H Kaplan. Analyzing tenant assignment policies. *Management science*, 33(3): 395–408, 1987.

[72] R. M Karp, U. V. Vazirani, and V. V. Vazirani. An optimal algorithm for on-line bipartite matching. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing (STOC)*, pages 352–358, 1990.

[73] Alexander Kleiner, Bernhard Nebel, and Vittorio A Ziparo. A mechanism for dynamic ride sharing based on parallel auctions. In *IJCAI*, volume 11, pages 266–272, 2011.

[74] Harold W Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics (NRL)*, 2(1-2):83–97, 1955.

[75] Benny Lehmann, Daniel Lehmann, and Noam Nisan. Combinatorial auctions with decreasing marginal utilities. *Games and Economic Behavior*, 55(2):270–296, 2006.

[76] J. Leshno. Dynamic Matching in Overloaded Waiting Lists. Working paper, 2014.

[77] Fei Li, Jay Sethuraman, and Clifford Stein. An optimal online algorithm for packet scheduling with agreeable deadlines. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 801–802. Society for Industrial and Applied Mathematics, 2005.

[78] Han Li, Mark D Stegall, Patrick G Dean, Edward T Casey, Kunam S Reddy, Hasan A Khamash, Raymond L Heilman, Martin L Mai, C Burcin Taner, Catherine L Kosberg, et al. Assessing the efficacy of kidney paired donationâĂŤperformance of an integrated three-site program. *Transplantation*, 98(3):300–305, 2014.

[79] Will Ma and David Simchi-Levi. Online resource allocation under arbitrary arrivals: Optimal algorithms and tight competitive ratios. 2017.

[80] Shafi Malik and Edward Cole. Foundations and principles of the canadian living donor paired exchange program. *Canadian journal of kidney health and disease*, 1(1):6, 2014.

[81] V. H. Manshadi, S. Oveis-Gharan, and A. Saberi. Online stochastic matching: online actions based on offline statistics. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1285–1294, 2011.

[82] Aranyak Mehta. Online matching and ad allocation. *Foundations and Trends® in Theoretical Computer Science*, 8(4):265–368, 2013.

[83] Aranyak Mehta, Amin Saberi, Umesh Vazirani, and Vijay Vazirani. Adwords and generalized online matching. *Journal of the ACM (JACM)*, 54(5):22, 2007.

[84] Herwig-Ulf Meier-Kriesche, Friedrich K Port, Akinlolu O Ojo, Steven M Rudich, Julie A Hanson, Diane M Cibrik, Alan B Leichtman, and Bruce Kaplan. Effect of waiting time on renal transplant outcome. *Kidney international*, 58(3):1311–1317, 2000.

[85] ML Melcher, DB Leeser, HA Gritsch, J Milner, S Kapur, S Busque, JP Roberts, S Katznelson, W Bry, H Yang, et al. Chain transplantation: initial experience of a large multicenter program. *American Journal of Transplantation*, 12(9):2429–2436, 2012.

[86] H. Mendelson. Market Behavior in a Clearing House. *Econometrica*, 50(6):1505–1524, 1982.

[87] Sean P Meyn and Richard L Tweedie. Stability of markovian processes iii: Foster–lyapunov criteria for continuous-time processes. *Advances in Applied Probability*, 25 (03):518–548, 1993.

[88] M. Molinaro and R. Ravi. Kidney Exchanges and the Query-commit Problem. Manuscript, 2013.

[89] Robert A Montgomery, Andrea A Zachary, Lloyd E Ratner, Dorry L Segev, Janet M Hiller, Julie Houp, Mathew Cooper, Louis Kavoussi, Thomas Jarrett, James Burdick, et al. Clinical results from transplanting incompatible live kidney donor/recipient pairs using kidney paired donation. *Jama*, 294(13):1655–1663, 2005.

[90] Robert A Montgomery, Sommer E Gentry, William H Marks, Daniel S Warren, Janet Hiller, Julie Houp, Andrea A Zachary, J Keith Melancon, Warren R Maley, Hamid Rabb, et al. Domino paired kidney donation: a strategy to make best use of live non-directed donation. *The Lancet*, 368(9533):419–421, 2006.

[91] Meguid El Nahas. The global challenge of chronic kidney disease. *Kidney international*, 68(6):2918–2929, 2005.

[92] Mohammadreza Nazari and Alexander L Stolyar. Reward maximization in general dynamic matching systems. *arXiv preprint arXiv:1608.01646*, 2016.

[93] A. Nikzad, M. Akbarpour, M. A. Rees, and A. E. Roth. Financing transplants' costs of the poor: A dynamic model of global kidney exchange. Technical report, Stanford University Working Paper, 2017.

[94] Michael Ostrovsky and Michael Schwarz. Carpooling and the economics of self-driving cars. Technical report, National Bureau of Economic Research, 2018.

[95] Erhun Ozkan and Amy R Ward. Dynamic matching for real-time ridesharing. 2016.

[96] K Park, SI Kim, JI Moon, and YS Kim. Exchange donor program in kidney transplantation. *Transplantation*, 65(5):161, 1998.

[97] Marco Pavone, Stephen L Smith, Emilio Frazzoli, and Daniela Rus. Robotic load balancing for mobility-ondemand systems. *Int J Rob Res*, 2012.

[98] Tomás Prieto-Rumeau and Onésimo Hernández-Lerma. Uniform ergodicity of continuous-time controlled markov chains: A survey and new results. *Annals of Operations Research*, 241(1):249–293, 2016. ISSN 1572-9338. doi: 10.1007/s10479-012-1184-4. URL `http://dx.doi.org/10.1007/s10479-012-1184-4`.

[99] Felix T Rapaport. The case for a living emotionally related international kidney donor exchange registry. In *Transplantation proceedings*, volume 18, page 5, 1986.

[100] M. A. Rees, J. E. Kopke, R. P. Pelletier, D. L. Segev, M. E. Rutter, A. J. Fabrega, J. Rogers, O. G. Pankewycz, J. Hiller, A. E. Roth, T. Sandholm, M. U. Ünver, and R. A. Montgomery. A non-simultaneous extended altruistic donor chain. *New England Journal of Medecine*, 360:1096–1101, 2009.

[101] Michael A Rees, Jonathan E Kopke, Ronald P Pelletier, Dorry L Segev, Matthew E Rutter, Alfredo J Fabrega, Jeffrey Rogers, Oleh G Pankewycz, Janet Hiller, Alvin E Roth, et al. A nonsimultaneous, extended, altruistic-donor chain. *New England Journal of Medicine*, 360(11):1096–1101, 2009.

[102] A. E. Roth. What have we learned from market design? *Economic Journal*, 118: 285–310, 2008.

[103] A. E. Roth, T. Sönmez, and M. U. Ünver. Kidney exchange. *Quarterly Journal of Economics*, 119:457–488, 2004.

[104] A. E. Roth, T. Sönmez, and M. U. Ünver. A kidney exchange clearinghouse in New England. *American Economic Review Papers and Proceedings*, 95(2):376–380, 2005.

[105] A. E. Roth, T. Sönmez, and M. U. Ünver. Pairwise kidney exchange. *Journal of Economic Theory*, 125:151–188, 2005.

[106] A. E. Roth, T. Sönmez, M. U. Ünver, F. L. Delmonico, and S. L. Saidman. Utilizing list exchange and nondirected donation through chain kidney paired donations. *American Journal of Transplantation*, 6:2694–2705, 2006.

[107] A. E. Roth, T. Sönmez, and M. U. Ünver. Efficient kidney exchange: coincidence of wants in markets with compatibility-based preferences. *American Economic Review*, 97:828–851, 2007.

[108] Alvin E Roth and Marilda Sotomayor. Two-sided matching. *Handbook of game theory with economic applications*, 1:485–541, 1992.

[109] Alvin E Roth, Tayfun Sönmez, M Utku Ünver, Francis L Delmonico, and Susan L Saidman. Utilizing list exchange and nondirected donation through 'chain' paired kidney donations. *American Journal of transplantation*, 6(11):2694–2705, 2006.

[110] Paat Rusmevichientong, Mika Sumida, and Huseyin Topaloglu. Dynamic assortment optimization for reusable products with random usage durations. Technical report, Working Paper, Cornell Tech, 2017.

[111] S. L. Saidman, A. E. Roth, T. Sönmez, M. U. Ünver, and F. L. Delmonico. Increasing the Opportunity of Live Kidney Donation by Matching for Two and Three Way Exchanges. *Transplantation*, 81:773–782, 2006.

[112] Susan L Saidman, Alvin E Roth, Tayfun Sönmez, M Utku Ünver, and Francis L Delmonico. Increasing the opportunity of live kidney donation by matching for two- and three-way exchanges. *Transplantation*, 81(5):773–782, 2006.

[113] Paolo Santi, Giovanni Resta, Michael Szell, Stanislav Sobolevsky, Steven H. Strogatz, and Carlo Ratti. Quantifying the benefits of vehicle pooling with shareability networks. In *Proc Natl Acad Sci USA*, 2014.

[114] Michael Schilde, Karl F Doerner, and Richard F Hartl. Metaheuristics for the dynamic stochastic dial-a-ride problem with expected return transports. *Computers & operations research*, 38(12):1719–1730, 2011.

[115] James Schummer. Influencing waiting lists. Technical report, Technical report, Working paper, Kellogg School of Management, 2015. 41, 2016.

[116] Dorry L Segev, Sommer E Gentry, J Keith Melancon, and Robert A Montgomery. Characterization of waiting times in a simulation of kidney paired donation. *American Journal of Transplantation*, 5(10):2448–2455, 2005.

[117] Dorry L Segev, Sommer E Gentry, Daniel S Warren, Brigitte Reeb, and Robert A Montgomery. Kidney paired donation and optimizing the use of live donor organs. *Jama*, 293(15):1883–1890, 2005.

[118] J Siegel-Itzkovich. Israel and Cyprus sign accord for sharing kidneys fromlive donors. `http://www.jpost.com/Israel-News/Israel-and-Cyprus-sign-accord-for-sharing-kidneys-from-live-donors-497422`, 2017. [Online].

[119] T. Sönmez, U. Ünver, et al. Market design for living-donor organ exchanges: An economic policy perspective. *Oxford Review of Economic Policy*, 33(4), 2017.

[120] Tayfun Sönmez and M Utku Ünver. Matching, allocation, and exchange of discrete resources. In *Handbook of social Economics*, volume 1, pages 781–852. Elsevier, 2011.

[121] Kevin Spieser, Samitha Samaranayake, Wolfgang Gruel, and Emilio Frazzoli. Shared-vehicle mobility-ondemand systems: A fleet operatorâĂŹs guide to rebalancing empty vehicles. In *Transportation Research Board 95th Annual Meeting*, 2016.

[122] Xuanming Su and Stefanos Zenios. Patient choice in kidney allocation: The role of the queueing discipline. *Manufacturing & Service Operations Management*, 6(4):280–301, 2004.

[123] M. Suthanthiran and T. B. Strom. Renal transplantation. *New England Journal of Medecine*, page 331:365, 1994.

[124] John N Tsitsiklis and Benjamin Van Roy. On average versus discounted reward temporal-difference learning. *Machine Learning*, 49(2-3):179–191, 2002.

[125] UNOS. United network for organ sharing. 2013. URL `http://www.unos.org/`.

[126] M. U. Ünver. Dynamic Kidney Exchange. *Review of Economic Studies*, 77(1):372–414, 2010.

[127] USRDS. United states renal data system. 2012. URL `http://www.usrds.org/qtr/default.aspx`.

[128] MM Vazifeh, P Santi, G Resta, SH Strogatz, and C Ratti. Addressing the minimum fleet problem in on-demand urban mobility. *Nature*, 557(7706):534, 2018.

[129] William Vickrey. Pricing as a tool in coordination of local transportation. In *Transportation economics*, pages 275–296. NBER, 1965.

[130] Hai Wang. Routing and scheduling for a last-mile transportation system. *Transportation Science*, 2017.

[131] Wikipedia contributors. Little's law., 2018. URL `https://en.wikipedia.org/wiki/Little%27s_law`. [Online; accessed 13-August-2018].

[132] Morteza Zadimoghaddam. Online weighted matching: Beating the $frac12$ barrier. *arXiv preprint arXiv:1704.05384*, 2017.

[133] S. A. Zenios. Optimal control of a paired-kidney exchange program. *Manage. Sci.*, 48(3):328–342, March 2002. ISSN 0025-1909. doi: 10.1287/mnsc.48.3.328.7732. URL `http://dx.doi.org/10.1287/mnsc.48.3.328.7732`.

[134] Rick Zhang and Marco Pavone. Control of robotic mobility-on-demand systems: a queueing-theoretical perspective. In *Proceedings of Robotics: Science and Systems Conference*, 2014.

Appendix A

Appendix of Chapter 4

A.1 Additional empirical results.

In Figure A-1 we show the distribution of edges and bilateral cycles in three cases. First we look at the real compatibility graph formed by agents in the NKR pool. Second, we form two model graphs while fitting the best parameters based on the data. Figure A-1 shows that a simple Erdős-Renyi model with the same number of edges has an edge-distribution that is very concentrated around the mean, contrary to the empirical graph where a significant number of agents have very few ingoing edges, and even fewer bilateral cycle options. While our 2-type model is still far from perfect, it captures some of this "sparse" behavior that is observed in the data.

A.2 Proof of Theorem 1

In order to study the Markov chains Π^b_τ and Π^G_τ, we define the following random variables:

- D^b_τ and D^G_τ represent the number of departures at the end of the τ'th batch, respectively for the *b-batching* and *greedy* policy.

- A_τ represents the number of H nodes during batch τ.

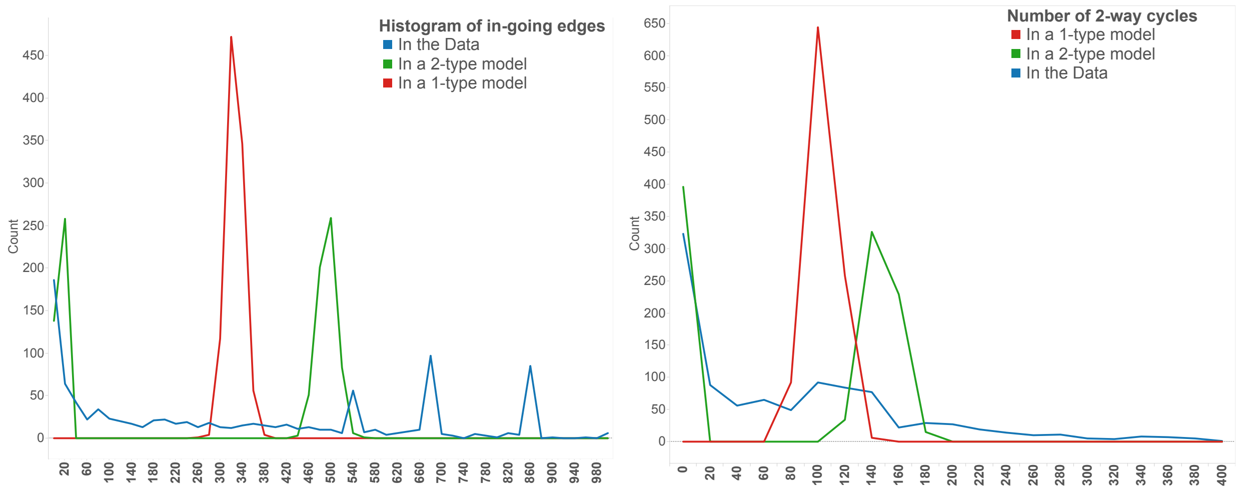

Figure A-1: These graphs show edge and cycle distributions for three types of graphs. In blue, the graph is derived from the clinical data with $n = 1000$ pairs that are not under-demanded. It has $d = 334780$ edges. In red, the graph is an instance of an Erdős-Renyi random graph with $n = 1000$ and $p = d/n^2 \approx 0.33$. In this instance it has $335,000$ edges. In green is an instance of a graph generated using a 2-type model, where node type is drawn from a bernoulli distribution with parameter $\rho = 0.4$. Edges to a node of type H (L) are drawn according to a bernoulli with parameter $p_H = 0.02$ ($p_E = 0.5$). In this instance it has $317,000$ edges. **Left:** distribution of the ingoing edges for three types of graphs. **Right:** distribution of the number of bilateral cycle options for all three types of graphs.

- Recall that $\mu_2^b(\tau)$ and $\mu_2^G(\tau)$ represent the number of H nodes matched in batch τ when only 2-cycles are allowed.

For the *b-batching* algorithm, using the notation defined we have: $\Pi_{\tau+1}^b = \Pi_\tau^b + A_\tau - D_\tau^b - \mu_2^b(\tau)$ Similarly for greedy: $\Pi_{t+1}^G = \Pi_\tau^G + A_\tau - D_\tau^G - \mu_2^G(\tau)$.
Let $\Delta_\tau = \Pi_\tau^b - \Pi_\tau^G$. We study $\mathbb{E}\left[\Delta_{t+1} | \Pi_\tau^b, \Pi_\tau^G\right]$.

$$\mathbb{E}\left[\Delta_{t+1} | \Pi_\tau^b, \Pi_\tau^G\right] = (1 - bq_H)\,\Delta_t - \mathbb{E}\left[\mu_2^b(\tau) - \mu_2^G(\tau) | \Pi_\tau^b, \Pi_\tau^G\right] \tag{A.1}$$

We will focus on the last term. Let $\widetilde{\mu}_2(\Pi_\tau^G)$ be the number of matches that would have been conducted, if the pool had started with Π_τ^G H nodes and had used *b-batching* at step τ.

$$\mathbb{E}\left[\mu_2^b(\tau) - \mu_2^G(\tau) | \Pi_\tau^b, \Pi_\tau^G\right] = \mathbb{E}\left[\mu_2^b(\tau) - \widetilde{\mu}_2(\Pi_\tau^G) | \Pi_\tau^b, \Pi_\tau^G\right] + \mathbb{E}\left[\widetilde{\mu}_2(\Pi_\tau^G) - \mu_2^G(\tau) | \Pi_\tau^b, \Pi_\tau^G\right]$$

The idea is to upper-bound $\mathbb{E}\left[\mu_2^b(\tau) - \widetilde{\mu}_2(\Pi_\tau^G) | \Pi_\tau^b, \Pi_\tau^G\right]$ and $\mathbb{E}\left[\widetilde{\mu}_2(\Pi_\tau^G) - \mu_2^G(\tau) | \Pi_\tau^b, \Pi_\tau^G\right]$ separately, and then prove bounds on $\mathbb{E}\left[\Delta_\infty\right]$.

Lemma 6. *Using only* 2*-ways, we have:* $\mathbb{E}\left[\widetilde{\mu}_2(\Pi_\tau^G) - \mu_2^G(\tau) | \Pi_\tau^b, \Pi_\tau^G\right] \leq \lambda_1 p_H$ *where* λ_1 *is defined in* (A.5)*.*

Proof. First ignore all the $L - L$ edges, and consider the bipartite graph between the arriving E nodes and all the H nodes. Let M^b and M^G be the matching formed respectively by *b-batching* and *greedy* in this graph. Note that based on the priorities set, *b-batching* finds a maximum matching in M^b. Consider $M^b \cup M^G$; this consists of disjoint (i) even-length paths/cycles (ii) odd-length paths with one more edge from M^b. Note that having an odd-length path that has more edges from M^G would contradict with M^b being maximum matching. In order to bound the difference between size of M^b and M^G, we bounds the number of odd-length paths.

First let us bound the number of such paths with length at least 3; Such a path would include at least one H node with degree at least 2. Therefore an upper-bound on the number of H nodes with degree at least 2 also serves as an upper bound on the number of these paths.

Consider a graph with N_H/N_E pairs of type H/E, and consider an H node v. We first compute the probability that node v has at least 2 neighbors among the E nodes.

$$\begin{aligned}
\mathbb{P}\left[\text{degree of } v \geq 2 | N_H, N_E\right] &= \sum_{j=2}^{N_E} \binom{N_E}{j} \left(p_H p_E\right)^j \left(1 - p_H p_E\right)^{N_E - j} \\
&= 1 - \left(1 - p_H p_E\right)^{N_E} - N_E p_H p_E \left(1 - p_H p_E\right)^{N_E - 1} \\
&\leq 1 - \left[1 - N_E p_H p_E\right] - \left[N_E p_H p_E - N_E (N_E - 1) \left(p_H p_E\right)^2\right] \\
&\leq N_E^2 \left(p_H p_E\right)^2, \qquad \text{(A.2)}
\end{aligned}$$

where in the second inequality we used the following Claim:

Claim 9. *Let $x \in [0, 1]$ and a positive integer i, then $(1 - x)^i \geq 1 - Yx$.*

Proof. The function $x \mapsto (1 - x)^i$ is convex. □

Next we take the expectation with respect to N_E and N_H. Given the pool size Π_τ^G, we have $N_H = \Pi_\tau^G - D_t^G + (b - N_E)$. Thus we have:

$$\begin{aligned}
&\mathbb{E}\left[\# \ H \text{ nodes with degree } \geq 2 | \Pi_\tau^b, \Pi_\tau^G\right] = \left(p_H p_E\right)^2 \mathbb{E}\left[\left(\Pi_\tau^G - D_t^G + (b - N_E)\right) N_E^2\right] \\
&\quad = \left(p_H p_E\right)^2 \left\{(1 - b q_H) \Pi_\tau^G \mathbb{E}\left[N_E^2\right] + \mathbb{E}\left[b N_E^2\right] - \mathbb{E}\left[N_E^3\right]\right\} \\
&\quad \leq \left(p_H p_E\right)^2 \left\{(1 - b q_H) \Pi_\tau^G (1 - \rho) b^2 + (1 - \rho) b^3 - (1 - \rho)^3 b^3\right\} \qquad \text{(A.3)}
\end{aligned}$$

where in the first equality we used the fact that $\mathbb{E}\left[D_\tau^G | \Pi_\tau^b, \Pi_\tau^G\right] = b q_H \Pi_\tau^G$, and that D_t^G and N_E are independent. In the first inequality we used Claim 10 and the fact that $0 \leq N_E \leq b$.

Claim 10. *Suppose $f(\cdot)$ is a monotone convex function and $x \in [l, r]$. We have: $f(\mathbb{E}[x]) \leq \mathbb{E}[f(x)] \leq \frac{f(r) - f(l)}{r - l} (\mathbb{E}[x] - l) + f(l)$.*

Proof. First inequality is just Jensen's. The second one follows from observing that $f(x) - f(l) \leq \frac{f(r)-f(l)}{r-l}(x-l)$, and taking expectation from both sides. □

Next, we bound the number of length-1 paths in $M^b \cup M^G$; such an edge can only exist if an E node arriving at a time i cannot give to any of the H nodes already in the pool, but can form a 2-way with an H node that arrives after i but before the end of the batch. Consider E node arriving at time $t = \tau b + i$; the expected number of 2-cycles it will form with an arriving H node is: $(b-i)\rho p_H p_E$. By Markov inequality, probability that node t forms at least one $H - L$ 2-cycle is at most $(b-i)\rho p_H p_E$.

$$\begin{aligned}\mathbb{E}\left[\#\text{ length-1 paths}\right] &\leq \sum_{i=1}^{b} \mathbb{P}\left[\text{node } t \text{ is } L\right] \mathbb{P}\left[t \text{ forms 2-way with } H \text{ node arriving after } t\right] \\ &\leq (1-\rho)\rho\frac{b^2}{2}p_H p_E. \end{aligned} \tag{A.4}$$

Putting (A.3) and (A.4), we get:

$$\lambda_1 = (1-\rho)\rho\frac{b^2}{2}p_E + p_E^2\left\{p_H(1-bq_H)\Pi_\tau^G(1-\rho)b^2 + p_H(1-\rho)b^3 - p_H(1-\rho)^3b^3\right\} \tag{A.5}$$

□

Lemma 7. *Using only 2-ways, we have:* $\mathbb{E}\left[\mu_2^b(\tau) - \widetilde{\mu}_2(\Pi_\tau^G)|\Pi_\tau^b, \Pi_\tau^G\right] \leq \lambda_2 p_H \left(\Pi_t^b - \Pi_\tau^G\right)^+$, *where λ_2 is defined in* (A.7).

Proof. If $\Pi_\tau^b \leq \Pi_\tau^G$ then if the same *b-batching* algorithm is used, coupling the random realizations allows us to always match more nodes in the pool that had more vertices at the start of the batch. Thus $\mu_2^b(\tau) \leq \widetilde{\mu}_2(\Pi_\tau^G)$

Now consider the case $\Pi_\tau^b \geq \Pi_\tau^G$. First ignore all the $L - L$ edges, and consider the bipartite graph between the arriving E nodes and all the H nodes in each pool. Suppose that the batch includes N_E pairs of type E. For simplicity of notation let $X = \Pi_\tau^G - D_t^G + (b - N_E)$

and $Y = \left(\Pi_\tau^b - D_t^b\right) - \left(\Pi_\tau^G - D_t^G\right)$. Consider a pool with $X + Y$ H nodes; partition the pool into two sets, set 1 with size X and set 2 with size Y. Consider an E node v; the probability that v cannot match set 1, but can match a node in set 2 is at most: $\left(1 - p_H p_E\right)^X \left[1 - \left(1 - p_H p_E\right)^Y\right]$. Therefore the expected size of maximum matching in the bipartite graph with $X+Y$ nodes is at most $N_E \left(1 - p_H p_E\right)^X \left[1 - \left(1 - p_H p_E\right)^Y\right]$ larger than the one with only X nodes.

$$\begin{aligned}
&\mathbb{E}\left[\mu_2^b(\tau) - \widetilde{\mu}_2(\Pi_\tau^G) | \Pi_\tau^b, \Pi_\tau^G, D_t^b, D_t^G\right] \qquad\qquad \text{(A.6)}\\
&\quad \leq \mathbb{E}\left[N_E \left(1 - p_H p_E\right)^{\Pi_\tau^G - D_t^G + (b - N_E)} \left[1 - \left(1 - p_H p_E\right)^{\left(\Pi_\tau^b - D_t\right) - \left(\Pi_\tau^G - D_t^G\right)}\right]\right]\\
&\quad = \left(1 - p_H p_E\right)^{\Pi_\tau^G - D_t^G + b} \left[1 - \left(1 - p_H p_E\right)^{\left(\Pi_\tau^b - D_t^b\right) - \left(\Pi_\tau^G - D_t^G\right)}\right] \mathbb{E}\left[N_E \left(1 - p_H p_E\right)^{-N_E}\right]\\
&\quad \leq \left(1 - p_H p_E\right)^{\Pi_\tau^G - D_t^G + b} \left[\left(\Pi_\tau^b - D_t\right) - \left(\Pi_\tau^G - D_t^G\right)\right] p_H p_E \mathbb{E}\left[N_E \left(1 - p_H p_E\right)^{-N_E}\right]\\
&\quad \leq \left(1 - p_H p_E\right)^{\Pi_\tau^G - D_t^G + b} \left[\left(\Pi_\tau^b - D_t^b\right) - \left(\Pi_\tau^G - D_t^G\right)\right] p_H p_E \left(1 - p_H p_E\right)^{-b} (1 - \rho) b\\
&\quad \leq \left[\left(\Pi_\tau^b - D_t^b\right) - \left(\Pi_\tau^G - D_t^G\right)\right] p_H p_E (1 - \rho) b
\end{aligned}$$

Where the second inequality uses the Claim 9, the third one holds by applying Claim 10 to the convex function $N_E \left(1 - p_H p_E\right)^{-N_E}$ for $0 \leq N_E \leq b$, and the last one holds because $\left(1 - p_H p_E\right)^{\Pi_\tau^G - D_t^G} \leq 1$. In order to complete the proof of Lemma 7, we take expectation with respect to D_t^b and D_t^G:

$$\mathbb{E}\left[\mu_2^b(\tau) - \widetilde{\mu}_2(\Pi_\tau^G) | \Pi_\tau^b, \Pi_\tau^G\right] \leq \left(1 - b q_H\right) \left(\Pi_\tau^b - \Pi_\tau^G\right) p_H p_E (1 - \rho) b,$$

and let:

$$\lambda_2 = (1 - b q_H) p_E (1 - \rho) b \tag{A.7}$$

□

Now note that by Claim 14, we have: $\left(\Pi_t^b - \Pi_\tau^G\right)^+ \leq 2b$. Combining this with Lemma 7, and applying it to (A.1) along with Lemma 6, we get: $\mathbb{E}\left[\Delta_{t+1} | \Pi_\tau^b, \Pi_\tau^G\right] \geq \left(1 - bq_H\right)\Delta_t - \lambda_1 p_H - 2\lambda_2 p_H b$. Taking expectation over Π_τ^b, Π_τ^G from both sides we get:

$$\mathbb{E}\left[\Delta_{t+1}\right] \geq \left(1 - bq_H\right)\mathbb{E}\left[\Delta_t\right] - \mathbb{E}\left[\lambda_1\right] p_H - 2\mathbb{E}\left[\lambda_2\right] p_H b \tag{A.8}$$

We now use the definition of λ_1 from (A.5), and take the expectation:

$$\mathbb{E}\left[\lambda_1\right] = (1-\rho)\rho\frac{b^2}{2}p_E + p_E^2\left\{p_H(1-bq_H)\mathbb{E}\left[\Pi_\tau^G\right](1-\rho)b^2 + p_H(1-\rho)b^3 - p_H(1-\rho)^3 b^3\right\}$$

Using auxiliary Lemma 10, we have: $\mathbb{E}\left[\Pi_\tau^G\right] \leq \rho/q_H$. This yields the following upper-bound on $\mathbb{E}\left[\lambda_1\right]$:

$$\bar{\lambda_1} = (1-\rho)\rho\frac{b^2}{2}p_E + p_E^2\left\{(1-bq_H)\rho(1-\rho)b^2/\alpha + p_H(1-\rho)b^3 - p_H(1-\rho)^3 b^3\right\}.$$

Substituting this in (A.8), we have:[1]

$$\mathbb{E}\left[\Delta_{t+1}\right] \geq \left(1 - bq_H\right)\mathbb{E}\left[\Delta_t\right] - \bar{\lambda_1} p_H - 2\bar{\lambda_2} p_H b$$

At $t = 0$, $\Delta_0 = 0$, therefore the above recursion implies:

$$\mathbb{E}\left[\Delta_\infty\right] \geq -\frac{p_H\left(\bar{\lambda_1} + 2\bar{\lambda_2} b\right)}{bq_H} \tag{A.9}$$

Now the expected match rate in the steady-state for the batching/greedy policy is $b\rho - bq_H\mathbb{E}\left[\Pi_\infty^b\right]$ and $b\rho - bq_H\mathbb{E}\left[\Pi_\infty^G\right]$, therefore (A.9) implies that:

$$\mathbb{E}\left[\mu_2^b(\infty)\right] \leq \mathbb{E}\left[\mu_2^G(\infty)\right] + p_H\lambda,$$

[1] Note that λ_2 is independent of Π_τ^b, Π_τ^G

where $\lambda = \bar{\lambda}_1 + 2\bar{\lambda}_2 b$, completing the proof.

A.3 3-way matching: proof of Theorem 2

Similar to the case for 2-way matching, we focus on the following recursion (As a reminder, $\Delta_\tau = \Pi^b_\tau - \Pi^G_\tau$).

$$\begin{aligned}
\mathbb{E}\left[\Delta_{t+1} | \Pi^b_\tau, \Pi^G_\tau\right] = {} & (1 - bq_H)\,\Delta_t \\
& - \mathbb{E}\left[\mu^b_3(\tau) - \widetilde{\mu}_3(\Pi^G_\tau) | \Pi^b_\tau, \Pi^G_\tau\right] \\
& - \mathbb{E}\left[\widetilde{\mu}_3(\Pi^G_\tau) - \mu^G_3(\tau) | \Pi^b_\tau, \Pi^G_\tau\right] \qquad \text{(A.10)}
\end{aligned}$$

Here we lower-bound $\mathbb{E}\left[\mu^b_3(\tau) - \widetilde{\mu}_3(\Pi^G_\tau) | \Pi^b_\tau, \Pi^G_\tau\right]$ and $\mathbb{E}\left[\widetilde{\mu}_3(\Pi^G_\tau) - \mu^G_3(\tau) | \Pi^b_\tau, \Pi^G_\tau\right]$ separately, and then prove an upper-bound on $\mathbb{E}\left[\Delta_\infty\right]$.

Lemma 8. *Using* 3*-ways, we have:* $\mathbb{E}\left[\widetilde{\mu}_3(\Pi^G_\tau) - \mu^G_3(\tau) | \Pi^b_\tau, \Pi^G_\tau\right] \geq \gamma_1 p_E^2$ *where* γ_1 *is given in* (A.11).

Proof. Suppose N_E pairs of type E arrive in the t-th batch. Also suppose all the H nodes arriving during the batch arrive at the very beginning.[2] Consider the greedy policy; each time an E-node arrives 3 cases can happen:

(i) The E-node can form a 2-way with an H-node in the pool. If this happens, we label this node at an A node.

(ii) The E-node cannot form a 2-way but forms at least one outgoing edge to an H-node. In this case we label the node as B node.

(iii) The arriving E nodes does not form any $L - H$ edge. We call these nodes C nodes.

Now let us consider how different type of E nodes match given the priorities set. An A node immediately matches an H node and leaves. The i-th B/C node matches the $i+1$-th B/C

[2] This can only help the greedy policy, and has no effect on the batching policy.

node if i is odd, and the $i-1$-th if even. We show that if the matching sequence includes a B_{2k-1}-B_{2k} pair and a $C_{2k'-1}$-$C_{2k'}$ pair, then with probability of at least p_E^2 we could have matched one more H node by waiting until the end of batch: consider 2 cases:

- Case 1: node B_{2k-1} and B_{2k} have outgoing edges to the same H node. In this case neither B_{2k-1} and B_{2k} will match each other in a 2-way. Now if we wait with probability of $2p_E - p_E^2 \geq p_E^2$ we can form a 3-way including B_{2k-1} or B_{2k} and $C_{2k'-1}$ or $C_{2k'}$.

- Case 2: node B_{2k-1} and B_{2k} have outgoing edges to different H nodes. In this case B_{2k-1} and B_{2k} can form at most one 3-way. However, if we also see $C_{2k'-1}$ and $C_{2k'}$ before making matching decisions, with probability of at least p_E^2 we can form 2 3-way matches. [3]

In the rest of the proof, we compute the probability that an E node gets a label of A,B, or C. We then we compute the probability of having B_{2k-1}-B_{2k} pairs and $C_{2k'-1}$-$C_{2k'}$ pairs. For simplicity of notation let $X = \Pi_\tau^G - D_t^G + (b - N_E)$ be the total number of H nodes in the pool. Probability E node being A is $p_A \triangleq \left[1-(1-p_H p_E)^X\right]$, being B is $p_B \triangleq (1-p_H p_E)^X - (1-p_H)^X$, and being C is $p_C \triangleq (1-p_H)^X$. Consider the first 4 E nodes. Probability of having B_1-B_2 and C_3-C_4 or the reverse is $2p_B^2 p_C^2$. Now considering the whole batch, in expectation, we have at least $0.5N_E p_B^2 p_C^2$ such subsequences. Therefore the gain of waiting is at least $0.5N_E p_B^2 p_C^2 p_E^2$. The next step is to take expectation with respect to N_E.

[3] The probability can be larger because node B_1 can have more than one outgoing $L->H$ edge.

$$
\begin{aligned}
&\mathbb{E}\left[\mu_3^G(\tau) - \widetilde{\mu}_3(\Pi_\tau^G) | \Pi_\tau^b, \Pi_\tau^G, D_\tau^G\right] \geq p_E^2/2\mathbb{E}\left[N_E p_B^2 p_C^2 | \Pi_\tau^b, \Pi_\tau^G, D_\tau^G\right] \\
&= \xi\mathbb{E}\left[N_E\left(1-\left(\frac{1-p_H}{1-p_Hp_E}\right)^X\right)^2(1-p_Hp_E)^{-2N_E}(1-p_H)^{-2N_E}\right] \\
&\geq \xi\left(1-\left(\frac{1-p_H}{1-p_Hp_E}\right)^{\Pi_\tau^G-D_\tau^G}\right)^2\mathbb{E}\left[N_E(1-p_Hp_E)^{-2N_E}(1-p_H)^{-2N_E}\right] \\
&\geq \xi\left(1-\left(\frac{1-p_H}{1-p_Hp_E}\right)^{\Pi_\tau^G-D_\tau^G}\right)^2(1-\rho)b(1-p_Hp_E)^{-2(1-\rho)Z}(1-p_H)^{-2(1-\rho)Z} \\
&\geq \frac{p_E^2}{2}(1-\rho)b(1-p_Hp_E)^{2(\rho\Pi_\tau^G+Z)}(1-p_H)^{2(\rho\Pi_\tau^G+Z)}\left(1-\left(\frac{1-p_H}{1-p_Hp_E}\right)^{\Pi_\tau^G}\right)^2\left[1-\frac{D_\tau^G}{\Pi_\tau^G}\right]^2
\end{aligned}
$$

where we used the shorthand $\xi = \frac{p_E^2}{2}(1-p_Hp_E)^{2(X+N_E)}(1-p_H)^{2(X+N_E)}$. The inequality in the third line follows from the fact that the function $\left(1-\left(\frac{1-p_H}{1-p_Hp_E}\right)^x\right)$ is decreasing in x. The inequality in the forth line follows form the Jensen's inequality. Finally in the last line we used the following:

$$
\frac{p_E^2}{2}(1-p_Hp_E)^{2(X+N_E)} \geq \frac{p_E^2}{2}(1-p_Hp_E)^{2(\Pi_\tau^G+b)},
$$

$$
\frac{p_E^2}{2}(1-p_H)^{2(X+N_E)} \geq \frac{p_E^2}{2}(1-p_H)^{2(\Pi_\tau^G+b)}, \text{and}
$$

$$
\left(1-\left(\frac{1-p_H}{1-p_Hp_E}\right)^{\Pi_\tau^G-D_\tau^G}\right)^2 \geq \left(1-\left(\frac{1-p_H}{1-p_Hp_E}\right)^{\Pi_\tau^G}\right)^2\left[1-\frac{D_\tau^G}{\Pi_\tau^G}\right]^2.
$$

The final step is to take exception with respect to D^G_τ:

$$\begin{aligned}\mathbb{E}\left[\mu^G_3(\tau) - \widetilde{\mu}^G_3(\tau)|\Pi^b_\tau, \Pi^G_\tau\right] &\geq \zeta \left(1 - \left(\frac{1-p_H}{1-p_Hp_E}\right)^{\Pi^G_\tau}\right)^2 \mathbb{E}\left[\left[1 - \frac{D^G_\tau}{\Pi^G_\tau}\right]^2\right] \\ &\geq \zeta \left(1 - \left(\frac{1-p_H}{1-p_Hp_E}\right)^{\Pi^G_\tau}\right)^2 (1-bq_H)^2,\end{aligned}$$

where we used the shorthand $\zeta = \frac{p_E^2}{2}(1-\rho)b(1-p_Hp_E)^{2(\rho\Pi^G_\tau+Z)}(1-p_H)^{2(\rho\Pi^G_\tau+Z)}$. The second inequality follows from Jensen's inequality and $\mathbb{E}\left[D^G_\tau||\Pi^b_\tau, \Pi^G_\tau\right] = bq_H\Pi^G_\tau$. We complete the proof by setting:

$$\gamma_1 = \zeta(1-bq_H)^2 \left(1 - \left(\frac{1-p_H}{1-p_Hp_E}\right)^{\Pi^G_\tau}\right)^2 \tag{A.11}$$

□

Lemma 9. *Using* 3*-ways, we have:* $\mathbb{E}\left[\mu^b_3(\tau) - \widetilde{\mu}_3(\Pi^G_\tau)|\Pi^b_\tau, \Pi^G_\tau\right] \geq -\gamma_2 p_H \left(\Pi^G_\tau - \Pi^b_\tau\right)^+$*, where* γ_2 *is defined in* (A.12)*.*

Proof. If $\Pi^b_\tau \geq \Pi^G_\tau$ then the result follows from a coupling argument similar to Lemma 7.

Now consider the case $\Pi^G_\tau > \Pi^b_\tau$. First ignore all the $L-L$ edges. Suppose the batch includes N_E pairs of type E. For simplicity of notation let $X = \Pi^b_\tau - D^b_t + (b - N_E)$ and $Y = (\Pi^G_\tau - D^G_t) - (\Pi^b_\tau - D^b_t)$. Consider a pool with $X+Y$ H nodes; partition the pool into two sets, set 1 with size X and set 2 with size Y. Let us first look at the graph formed between the N_E pairs of type E and set 1. Similar to the proof of Lemma 8, we label an E node $A_1/B_1/C_1$ if it forms at least one 2-way/no 2-way but at least one outgoing edge/no outgoing edge. We can define a similar labeling of $A_2/B_2/C_2$ in the graph that includes both set 1 and set 2. We claim that $\mu^G_3(\tau) - \mu^b_3(\tau) \leq |C_1| - |C_2|$, where we abuse the notation of C_1/C_2 to also denote set of E nodes with such labels. This follows from the observation that if an E node is labeled C_1 but not C_2 it gets the label of either A_2 or B_2. In either case it can result in matching at most one more H node. The next step involves taking the expectation with respect to the labeling and N_E.

$$\begin{aligned}
&\mathbb{E}\left[\mu(\Pi_\tau^G) - \mu(\Pi_\tau^b)|\Pi_\tau^b, \Pi_\tau^G, D_t^b, D_t^G\right] \le \mathbb{E}\left[|C_1| - |C_2||\Pi_\tau^b, \Pi_\tau^G, D_t^b, D_t^G\right] \\
&= \mathbb{E}\left[N_E\left((1-p_H)^X - (1-p_H)^{X+Y}\right)|\Pi_\tau^b, \Pi_\tau^G, D_t^b, D_t^G\right] \\
&= (1-p_H)^{\Pi_\tau^b + b}\mathbb{E}\left[N_E(1-p_H)^{-N_E}\left(1-(1-p_H)^Y\right)|\Pi_\tau^b, \Pi_\tau^G, D_t^b, D_t^G\right] \\
&\le (1-p_H)^{\Pi_\tau^b + b}\left[(\Pi_\tau^G - D_t^G) - (\Pi_\tau^b - D_t^b)\right]p_H\mathbb{E}\left[N_E(1-p_H)^{-N_E}|\Pi_\tau^b, \Pi_\tau^G, D_t^b, D_t^G\right] \\
&\le (1-p_H)^{\Pi_\tau^b + b}\left[(\Pi_\tau^G - D_t^G) - (\Pi_\tau^b - D_t^b)\right]p_H(1-p_H)^{-b}(1-\rho)b \\
&\le \left[(\Pi_\tau^G - D_t^G) - (\Pi_\tau^b - D_t^b)\right]p_H(1-\rho)b
\end{aligned}$$

where the inequality in the forth line follows from Claim 9, and the one in the fifth line follows from Claim 10 in a similar way we we did in the proof of Lemma 7. The last equality follows from $(1-p_H)^{\Pi_\tau^b} \le 1$. To complete the proof, we take expectation with respect to D_t^b and D_t^G, and let:

$$\gamma_2 = (1-\rho)(1-bq_H)b. \tag{A.12}$$

□

Auxiliary Claim 14 implies that $\left(\Pi_\tau^G - \Pi_t^b\right)^+ \le \left(\Pi_\tau^G - \Pi_t^b\right) + 2b$. Combining this with Lemma 9, and also applying Lemma 8 to (A.10), we get:

$$\mathbb{E}\left[\Delta_{t+1}|\Pi_\tau^b, \Pi_\tau^G\right] \le (1-bq_H)\,\Delta_t - \gamma_2 p_H \Delta_t + 2\gamma_2 p_H b - \gamma_1 p_E^2$$

Taking expectation over Π_τ^b, Π_τ^G from both sides we get:

$$\mathbb{E}\left[\Delta_{t+1}\right] \leq \left[1 - bq_H - \gamma_2 p_H\right] \mathbb{E}\left[\Delta_t\right] + 2\gamma_2 p_H b - \mathbb{E}\left[\gamma_1\right] p_E^2 \tag{A.13}$$

Note that γ_2 given in (A.12) is a constant independent of Π_τ^b, Π_τ^G. Claim 11 provides a lower-bound on $\mathbb{E}\left[\gamma_1\right]$. Together with recursion (A.13) we get:

$$\mathbb{E}\left[\Delta_\infty\right] \leq -\frac{\gamma_1 p_E^2 - 2\gamma_2 Z p_H}{bq_H + \gamma_2 p_H} \tag{A.14}$$

The expected match rate in the steady-state for the *b-batching* policy is $\mathbb{E}\left[\mu_3^b(\infty)\right] = b\rho - bq_H \mathbb{E}\left[\Pi_\infty^b\right]$ and for the *greedy* policy: $\mathbb{E}\left[\mu_3^G(\infty)\right] = b\rho - bq_H \mathbb{E}\left[\Pi_\infty^G\right]$. Therefore (A.14) implies that:

$$\mathbb{E}\left[\mu_3^b(\infty)\right] \geq \mathbb{E}\left[\mu_3^G(\infty)\right] + \frac{\left(\gamma_1 p_E^2 - 2b^2(1-\rho)(1-bq_H)p_H\right)\alpha}{\alpha + (1-\rho)(1-bq_H)},$$

A.4 Auxiliary claims

Claim 11. *Suppose* $\epsilon = \max\left\{\sqrt{\frac{6q_H}{0.5\rho}}, \sqrt{\frac{4q_H[1+(1-\rho)/\alpha]}{0.5\rho(1-p_H p_E)}}\right\}$*, and parameters* ρ*,* p_H*,* q_H*, and* p_E *are such that* $\epsilon < 1$*. For* $t > \max\{\frac{\log 0.5}{\log(1-bq_H)}, \frac{\log 0.5}{Z \log\left[(1-q_H)(\rho+(1-\rho)(1-p_H))\right]}\}$*, we have:*

$$\begin{aligned} \mathbb{E}\left[\gamma_1\right] &\geq \gamma_1 \\ &\triangleq \eta \left(\frac{1-p_E}{2(1-p_H)}\right)^2 \left[0.72 \min_{x\in[2(1-\epsilon)\widetilde{l}\rho/\alpha, 2(1+\epsilon)\rho/\alpha]} xe^{-x}\right]^2 \end{aligned} \tag{A.15}$$

where $\widetilde{l} = \frac{0.5(1-p_H p_E)}{1+(1-\rho)/\alpha}$ *as defined in Lemma 12, and* $\eta = \frac{1}{2}(1-bq_H)^2(1-p_H p_E)^{2\rho Z}(1-p_H)^{2\rho Z}$*.*

Proof.

$$\begin{aligned}\mathbb{E}\left[\gamma_1\right] &= \eta\mathbb{E}\left[\left((1-p_Hp_E)^{\Pi_\tau^G}(1-p_H)^{\Pi_\tau^G}\left(1-\left(\frac{1-p_H}{1-p_Hp_E}\right)^{\Pi_\tau^G}\right)\right)^2\right] \\ &\geq \eta\mathbb{E}\left[(1-p_Hp_E)^{\Pi_\tau^G}(1-p_H)^{\Pi_\tau^G}\left(1-\left(\frac{1-p_H}{1-p_Hp_E}\right)^{\Pi_\tau^G}\right)\right]^2 \qquad \text{(A.16)}\end{aligned}$$

where the inequality in the second line is by Jensen's inequality. In the rest we focus on the $\mathbb{E}\left[(1-p_Hp_E)^{\Pi_\tau^G}(1-p_H)^{\Pi_\tau^G}\left(1-\left(\frac{1-p_H}{1-p_Hp_E}\right)^{\Pi_\tau^G}\right)\right]$ and provide a lower bound for this term.

$$\begin{aligned}&(1-p_Hp_E)^{\Pi_\tau^G}(1-p_H)^{\Pi_\tau^G}\left(1-\left(\frac{1-p_H}{1-p_Hp_E}\right)^{\Pi_\tau^G}\right) \\ &\qquad = (1-p_H)^{\Pi_\tau^G}\left((1-p_Hp_E)^{\Pi_\tau^G}-(1-p_H)^{\Pi_\tau^G}\right) \\ &\qquad \geq (1-p_H)^{2\Pi_\tau^G-1}\Pi_\tau^G\left(1-p_E\right)p_H \\ &\qquad \geq \frac{1-p_E}{2(1-p_H)}2p_H\Pi_\tau^G e^{-2p_H\Pi_\tau^G}\end{aligned}$$

where the inequality in the second line follows form applying Claim 12, and the inequality in the last line follows from $\log(1-p_H) \geq -p_H$, multiplying both sides by $2\Pi_\tau^G$ and taking exponential from both sides. Let ϵ be a small positive number, $\mathbb{I}\cdot$ be the indicator function, and $\widetilde{l} < 1$ be the constant defined in Lemma 12.

$$\begin{aligned}&\mathbb{E}\left[2p_H\Pi_\tau^G e^{-2p_H\Pi_\tau^G}\right] \geq \mathbb{E}\left[2p_H\Pi_\tau^G e^{-2p_H\Pi_\tau^G}\mathbb{I}(1-\epsilon)\widetilde{l}\rho/q_H \leq \Pi_\tau^G \leq (1+\epsilon)\rho/q_H\right] \\ &\geq \mathbb{P}\left[(1-\epsilon)\widetilde{l}\rho/q_H \leq \Pi_\tau^G \leq (1+\epsilon)\rho/q_H\right]\min_{x\in[2(1-\epsilon)\widetilde{l}\rho/\alpha,2(1+\epsilon)\rho/\alpha]} xe^{-x} \\ &= \left(1-\mathbb{P}\left[\Pi_\tau^G < (1-\epsilon)\widetilde{l}\rho/q_H\right]-\mathbb{P}\left[\Pi_\tau^G > (1+\epsilon)\rho/q_H\right]\right)\min_{x\in[2(1-\epsilon)\widetilde{l}\rho/\alpha,2(1+\epsilon)\rho/\alpha]} xe^{-x}\end{aligned}$$

Using the concentration results of Lemmas 12 and 10 we have for

$$t > \max\left\{\frac{\log 0.5}{\log(1-bq_H)}, \frac{\log 0.5}{b\log\left[(1-q_H)\left(\rho+(1-\rho)(1-p_H)\right)\right]}\right\}$$

$$\mathbb{E}\left[2p_H\Pi_\tau^G e^{-2p_H\Pi_\tau^G}\right] \geq \left(1 - e^{-\frac{0.5\epsilon^2\rho}{3q_H}} - e^{-\frac{0.5\rho(1-p_Hp_E)\epsilon^2}{2q_H[1+(1-\rho)/\alpha]}}\right)\min_{x\in[2(1-\epsilon)\widetilde{l}\rho/\alpha, 2(1+\epsilon)\rho/\alpha]} xe^{-x}.$$

Substituting this back to (A.16), we have:

$$\mathbb{E}\left[\gamma_1\right] \geq \eta\left(\frac{1-p_E}{2(1-p_H)}\right)^2\left[\left(1 - e^{-\frac{0.5\epsilon^2\rho}{3q_H}} - e^{-\frac{0.5\rho(1-p_Hp_E)\epsilon^2}{2q_H[1+(1-\rho)/\alpha]}}\right)\min_{x\in[2(1-\epsilon)\widetilde{l}\rho/\alpha, 2(1+\epsilon)\rho/\alpha]} xe^{-x}\right]^2 \tag{A.17}$$

Finally, it follows form the definition of ϵ that $e^{-\frac{0.5\epsilon^2\rho}{3q_H}} \leq e^{-2} = 0.1353$. Similarly, $e^{-\frac{0.5\rho(1-p_Hp_E)\epsilon^2}{2q_H[1+(1-\rho)/\alpha]}} \leq e^{-2} = 0.1353$. Substituting this in (A.17), we get:

$$\mathbb{E}\left[\gamma_1\right] \geq \eta\left(\frac{1-p_E}{2(1-p_H)}\right)^2\left[0.72\min_{x\in[2(1-\epsilon)\widetilde{l}\rho/\alpha, 2(1+\epsilon)\rho/\alpha]} xe^{-x}\right]^2$$

□

Claim 12. *For $Y > 1$ and $0 \leq x_1 \leq x_2 \leq 1$, we have: $(1-x_1)^Y - (1-x_2)^Y \geq Y(1-x_2)^{Y-1}(x_2-x_1)$.*

Proof. Consider the function $f(x) = (1-x)^Y$, and consider $0 \leq x_1 \leq x_2 \leq 1$. By mean-value theorem, there exists $\xi \in [x_1, x_2]$ such that:

$$f(x_2) - f(x_1) = f'(\xi)(x_2 - x_1)$$

Function $f(\cdot)$ is convex and decreasing in $[0,1]$. This implies $f'(\xi) \leq f'(x_2) = -Y(1-x_2)^{Y-1}$.

Therefore,

$$f(x_1) - f(x_2) \geq Y(1-x_2)^{Y-1}(x_2 - x_1).$$

□

Claim 13. *At any batch step t and for both 2-way and 2,3-way matching , we have:*

- *if* $\Pi_\tau^G - \Pi_t^b \geq 0$ *then* $\Pi_\tau^G - \Pi_t^b \geq D_t^G - D_t^b \geq 0$.
- *if* $\Pi_\tau^G - \Pi_t^b \leq 0$ *then* $\Pi_\tau^G - \Pi_t^b \leq D_t^G - D_t^b \leq 0$.

Proof. For each agent i in the pool at τ, let X_i^b and X_i^G be the random variable indicating that i departs at τ under the *b-batching* and *greedy* policies. Clearly, X_i^b is a Bernoulli random variable with mean bq_H.

Suppose that $\Pi_\tau^G \leq \Pi_t^b$. For each agent i in the batching pool, that is not in the greedy pool, we assume that $X_i^G = 0$, which allows us to couple all the variables X_i^b and X_i^G such that their joint realizations satisfy $X_i^b \geq X_i^G$. This yields the result:

$$D_t^G = \sum_{i=0}^{\Pi_\tau^G} X_i^G = \sum_{i=0}^{\Pi_\tau^b} X_i^G \leq \sum_{i=0}^{\Pi_\tau^b} X_i^b = D_t$$

The proof in the case $\Pi_\tau^G \geq \Pi_t^b$ follows from the same argument. □

Claim 14. *At any batch step t and for both 2-way and 2,3-way matching , we have $\Pi_\tau^G \geq \Pi_t^b - 2b$*

Proof. As a reminder, let $\Delta_\tau = \Pi_\tau^b - \Pi_\tau^G$. We first prove that $\Delta_t \geq 0 \Rightarrow \Delta_{t+1} \leq \Delta_t$. Recall that:

$$\Delta_{t+1} - \Delta_\tau = A_\tau - A_\tau - D_\tau^b + D_\tau^G - \mu_2^b(\tau) + \mu_2^G(\tau). \tag{A.18}$$

We now couple arrivals such that for all t, $A_\tau = A_\tau$. Using the assumption $\Delta_t \geq 0$, and Claim 13, we get $D_\tau^b - D_\tau^G \geq 0$.

We now couple the compatibility realizations between the arriving nodes and the H nodes present in the pool. Because there are more nodes in the batching pool (even after accounting for departures, cf Claim 13), for each match that is conducted in the greedy pool, we can associate the corresponding match in the batching pool. Therefore the matches conducted by greedy are a feasible solution for the batching optimization, and $\mu_2^b(\tau) \geq \mu_2^G(\tau)$. This concludes that $\Delta_t \geq 0 \Rightarrow \Delta_{t+1} \leq \Delta_t$.

Suppose now that $\Delta_t \leq 0$. Then from Claim 13, $\Delta_t - D_\tau^b + D_\tau^G \leq 0$, and using Equation A.18, we get $\Delta_{t+1} \leq \mu_2^G(\tau) \leq 2b$, where the last inequality comes from the fact that each incoming node can never match more than two other nodes. This concludes that for all t, $\Delta_t \leq 2b$. □

Lemma 10. *For both* 2*-way and* 2, 3*-way matching*

1. *At any batch step* $\tau > 0$*, we have* $\mathbb{E}\left[\Pi_\tau^G\right] \leq \rho/q_H$ *and* $\mathbb{E}\left[\Pi_\tau^b\right] \leq \rho/q_H$.
2. *At any batch step* $\tau > \frac{\log 0.5}{\log(1-bq_H)}$*, for a small constant* ϵ*, we have:*

$$\mathbb{P}\left[\Pi_\tau^G \geq (1+\epsilon)\rho/q_H\right] \leq e^{-\frac{0.5\epsilon^2\rho}{3q_H}}.$$

The same holds for Π_τ^b

Proof. We compute the pool size of a "hypothetical process" where nodes do not leave the pool after getting matched, i.e., they only leave according to the probabilistic departure process. This gives us an upper-bound on the pool size of the actual system. Let $X_{t,\tau}$ be the indicator that node arriving at time $t < \tau b$ is still in the pool at the beginning of the τ-th batch in the "hypothetical process". For $1 \leq t \leq \tau b$, the random variables $X_{t,\tau}$ are independent Bernoulli with success probability of $\rho(1-bq_H)^{\lfloor \frac{\tau b - t}{b} \rfloor}$. Thus we have $\Pi_\tau^b \leq \sum_{t=1}^{\tau b} X_{t,\tau}$. Taking expectation we get:

$$\begin{aligned}\mathbb{E}\left[\Pi_\tau^G\right] \leq \sum_{t=1}^{\tau b} \mathbb{E}\left[X_{t,\tau}\right] &\leq \sum_{t=1}^{\tau b} \left[\rho(1-bq_H)^{\lfloor \frac{\tau b - t}{b} \rfloor}\right] \\ &\leq \sum_{j=0}^{\infty} \left[\rho b(1-bq_H)^j\right] \leq \rho/q_H.\end{aligned}$$

Further using a concentration result due to [10] for independent Bernoulli variables, we have:

$$\begin{aligned}\mathbb{P}\left[\Pi_\tau^G \geq (1+\epsilon)\rho/q_H\right] &\leq \mathbb{P}\left[\sum_{t=1}^{\tau b} X_{t,\tau} \geq (1+\epsilon)\rho/q_H\right] \\ &\leq \mathbb{P}\left[\sum_{t=1}^{\tau b} X_{t,\tau} \geq (1+\epsilon)\sum_{t=1}^{\tau b}\mathbb{E}\left[X_{t,\tau}\right]\right] \leq e^{-\frac{\epsilon^2\sum_{t=1}^{\tau b}\mathbb{E}[X_{t,\tau}]}{3}} \leq e^{-\frac{0.5\epsilon^2\rho}{3q_H}},\end{aligned}$$

where the last inequality follows from $\sum_{t=1}^{\tau b}\mathbb{E}\left[X_{t,\tau}\right] \geq 0.5\rho/q_H$ for $t > \frac{\log 0.5}{b\log(1-q_H)}$.

□

Lemma 11. *For any batch step* $t \geq \frac{\log 0.5}{\log\left[(1-bq_H)(1-p_H)^{b(1-\rho)}\right]}$*, and for both* $k=2$ *and* $k=3$*, we have* $\mathbb{E}\left[\Pi_t^b\right] \geq \frac{l\rho}{q_H}$ *where* $l = \frac{0.5(1-(1-\rho)p_H b)}{1+(1-\rho)/\alpha}$.

Proof. We prove this lower bound by counting the number of H nodes with degree zero during their time in the pool. Consider node v joining the pool during the τ'-th batch where $\tau' < \tau$. If $N_{L,\tau'}$ nodes arrive during the τ'-th batching, then node v has $(1-p_H)^{N_{L,\tau'}}$ probability of not receiving an edge from E nodes. Now consider any $\tau' < j \leq \tau$, with probability at least $(1-bq_H)^{j-\tau'}(1-p_H)^{N_{L,\tau'}+N_{L,\tau'+1}+\ldots+N_{L,j}}$ node v will still be in the pool without forming any incoming edge. Let Z be the set of nodes from the τ'-th batch still in pool at end of the τ-th batch which had no chance of matching. We have:

$$\begin{aligned}&\mathbb{P}\left[v \in Z | v \text{ is } H,\ N_{L,\tau},\ N_{L,\tau'+1},\ldots,N_{L,\tau}\right] \\ &\qquad \geq (1-bq_H)^{\tau-\tau'}(1-p_H)^{N_{L,\tau'}+N_{L,\tau'+1}+\ldots+N_{L,\tau}}\end{aligned}$$

Summing over all the H nodes in the τ'-th batch, and taking expectation with respect to $N_{L,\tau'}, N_{L,\tau'+1},\ldots N_{L,\tau}$ we get:

$$
\begin{aligned}
\mathbb{E}\left[|Z|\right] &\geq (1-bq_H)^{\tau-\tau'}\mathbb{E}\left[(Z-N_{L,\tau'})(1-p_H)^{N_{L,\tau'}+N_{L,\tau'+1}+\ldots+N_{L,\tau}}\right] \\
&\geq (1-bq_H)^{\tau-\tau'}\mathbb{E}\left[(Z-N_{L,\tau'})(1-p_H)^{N_{L,\tau'}}\right]\mathbb{E}\left[(1-p_H)^{N_{L,\tau'+1}}\right]\ldots\mathbb{E}\left[(1-p_H)^{N_{L,\tau}}\right] \\
&\geq (1-bq_H)^{\tau-\tau'}\mathbb{E}\left[(Z-N_{L,\tau'})(1-p_H N_{L,\tau'})\right]\mathbb{E}\left[(1-p_H)^{N_{L,\tau'+1}}\right]\ldots\mathbb{E}\left[(1-p_H)^{N_{L,\tau}}\right] \\
&\geq (1-bq_H)^{\tau-\tau'}\rho b(1-(1-\rho)p_H b)(1-p_H)^{(1-\rho)b(t-\tau'-1)}
\end{aligned}
$$

where the inequality in the forth line is by Claim 9 and the the one in the last line is by Jensens's equality. Summing over all such v nodes in batches $\tau' < \tau$ gives us a lower bound on the on pool size at the end of the τ-th batch.

$$
\begin{aligned}
\mathbb{E}\left[\Pi_\tau^b\right] &\geq \rho b(1-(1-\rho)p_H b)\sum_{\tau'=1}^{\tau}\left[(1-bq_H)(1-p_H)^{b(1-\rho)}\right]^{\tau-\tau'} \\
&= \rho b(1-(1-\rho)p_H b)\frac{1-\left[(1-bq_H)(1-p_H)^{b(1-\rho)}\right]^{\tau}}{1-\left[(1-bq_H)(1-p_H)^{b(1-\rho)}\right]} \\
&\geq \frac{0.5\rho b(1-(1-\rho)p_H b)}{1-\left[(1-bq_H)(1-p_H)^{b(1-\rho)}\right]} \\
&\geq \frac{0.5\rho b(1-(1-\rho)p_H b)}{bq_H(1+(1-\rho)/\alpha)} \\
&= \frac{\rho}{q_H}\frac{0.5(1-(1-\rho)p_H b)}{1+(1-\rho)/\alpha}
\end{aligned}
$$

where inequality in the third line follows from the assumption that $\tau \geq \frac{\log 0.5}{\log\left[(1-bq_H)(1-p_H)^{b(1-\rho)}\right]}$, and the one in the forth line follows from $\left[(1-bq_H)(1-p_H)^{b(1-\rho)}\right] \geq 1-bq_H(1+(1-\rho)/\alpha)$.

□

Lemma 12. *For any batch step $\tau \geq \frac{\log 0.5}{b\log\left[(1-q_H)(\rho+(1-\rho)(1-p_H))\right]}$ and for both $k=2$ and $k=3$,*

1. $\mathbb{E}\left[\Pi_\tau^G\right] \geq \frac{\widetilde{l}\rho}{q_H}$, *where* $\widetilde{l} = \frac{0.5(1-p_H p_E)}{1+(1-\rho)/\alpha}$.

2. *For a small constant ϵ, we have:* $\mathbb{P}\left[\Pi_\tau^G \leq (1-\epsilon)\widetilde{l}\rho/q_H\right] \leq e^{-\frac{0.5\rho(1-p_H p_E)\epsilon^2}{2q_H[1+(1-\rho)/\alpha]}}$.

Proof. We prove this lower bound by counting the number of H nodes with degree zero during their time in the pool. Consider node v joining the pool at time $t < \tau b$. Assume at each time there is exactly one E node waiting to be matched. [4] Node v can form a 2-way with the waiting E node independently with probability $p_H p_E$. If this event does not happen, then the event that node v does not leave then pool by time τb and also does not form any incoming edge with an arriving node between time t and τb is independent of all the other events and it has probability $(1-bq_H)^{\left\lfloor \frac{\tau b - t}{b} \right\rfloor} (\rho + (1-\rho)(1-p_H))^{\tau b - t}$. Let $X_{t,\tau b}$ be the indicator that the first event does not happen for node v, but the second one does. It follows by definition that $\Pi^G_\tau \geq \sum_{t=1}^{\tau b} X_{t,\tau b}$. Taking expectation form both sides we get:

$$
\begin{aligned}
\mathbb{E}\left[\Pi^G_\tau\right] &\geq \sum_{t=1}^{\tau b} \mathbb{E}\left[X_{t,\tau b}\right] = \rho\left(1 - p_H p_E\right) \sum_{t=1}^{\tau b} (1-bq_H)^{\left\lfloor \frac{\tau b - t}{b} \right\rfloor} \left(\rho + (1-\rho)(1-p_H)\right)^{\tau b - t} \\
&\geq \rho\left(1 - p_H p_E\right) \sum_{t=1}^{\tau b} \left[(1-q_H)\left(\rho + (1-\rho)(1-p_H)\right)\right]^{\tau b - t} \\
&= \rho\left(1 - p_H p_E\right) \frac{1 - \left[(1-q_H)\left(\rho + (1-\rho)(1-p_H)\right)\right]^{\tau b}}{1 - \left[(1-q_H)\left(\rho + (1-\rho)(1-p_H)\right)\right]} \\
&\geq \frac{0.5\rho\left(1 - p_H p_E\right)}{q_H\left[1 + (1-\rho)/\alpha\right]},
\end{aligned}
$$

where the inequality in the second line holds because $(1-bq_H)^{\left\lfloor \frac{\tau b - t}{b} \right\rfloor} \geq (1-bq_H)^{\frac{\tau b - t}{b}}$ and by Claim 9 $(1-bq_H)^{\frac{\tau b - t}{b}} \geq (1-q_H)^{\tau b - t}$. Further using a concentration for independent Bernoulli variables [10], we have:

$$
\begin{aligned}
\mathbb{P}\left[\Pi^G_\tau \leq (1-\epsilon)\widetilde{l}\rho/q_H\right] &\leq \mathbb{P}\left[\sum_{t=1}^{\tau b} X_{t,\tau} \leq (1-\epsilon)\widetilde{l}\rho/q_H\right] \\
&\leq \mathbb{P}\left[\sum_{t=1}^{\tau b} X_{t,\tau} \geq (1-\epsilon)\sum_{t=1}^{\tau b} \mathbb{E}\left[X_{t,\tau}\right]\right] \\
&\leq e^{-\frac{\epsilon^2 \sum_{t=1}^{\tau b} \mathbb{E}[X_{t,\tau}]}{2}} \leq e^{-\frac{0.5\rho(1-p_H p_E)\epsilon^2}{2q_H[1+(1-\rho)/\alpha]}},
\end{aligned}
$$

[4] In fact, there is at most one E node waiting, but this can only decrease the pool size.

□

Appendix B

Appendices for Chap 5

B.1 Proof of Lemma 2

Proof of Lemma 2. Let $\pi(x,y)$ be the state joint distribution of $[X,Y]$, and let $\pi_X(x) = \sum_{y\geq 0}\pi(x,y)$ be the marginal distribution of X. For a given $x \geq 0$, we have:

$$\sum_{y\in S(x)}\pi(x,y)Q([x,y],[x+1,y]) + \sum_{y\notin S(x)}\pi(x,y)Q([x,y],[x+1,y]) \\ = \sum_{y\in S(x+1)}\pi(x+1,y)Q([x+1,y],[x,y]) + \sum_{y\notin S(x+1)}\pi(x+1,y)Q([x+1,y],[x,y]).$$

Using Condition 2, we upper-bound the LHS and lower-bound the RHS, which results in having:

$$f(x)\mathbb{P}[X=x, Y\in S(x)] + \mathbb{P}[X=x, Y\notin S(x)] \geq g(x+1)\mathbb{P}[X=x+1, Y\in S(x+1)].$$

Let $\pi_S(x) = \mathbb{P}[X=x, Y\in S(x)] = \sum_{y\in S(x)}\pi(x,y)$. Observe that $\pi_X(x) \leq \pi_S(x) + c\delta^x \leq \pi_S(x) + c\rho^x$ from Condition 1 and for any $\rho \in [\delta, 1)$. Assuming that for $x \geq \eta$, $\frac{f(x)}{g(x+1)} \leq \rho$, we get:

$$\pi_S(x+1) \leq \frac{f(x)}{g(x+1)}\pi_S(x) + \frac{\mathbb{P}[Y\notin S(x)]}{g(x+1)} \leq \rho\pi_S(x) + \frac{c\delta^x}{g(x+1)} \leq \rho\pi_S(x) + \frac{c\rho^x}{g(\eta+1)},$$

where the last inequality results from the assumption $\frac{\delta^x}{g(x+1)} \le \frac{\rho^x}{g(\eta+1)}$. We can now prove by induction that:

$$\pi_S(\eta+i) \le \rho^i \left(\pi_S(\eta) + i \frac{c\rho^{\eta-1}}{g(\eta+1)} \right)$$

This allows us to conclude:

$$\begin{aligned}
\mathbb{P}[X \ge \eta + k] &= \sum_{i=\eta+k}^{\infty} \pi_X(i) \\
&\le \sum_{i=k}^{\infty} \pi_S(\eta+i) + \sum_{i=k}^{\infty} c\rho^{\eta+i} \\
&\le \sum_{i=k}^{\infty} \rho^i \pi_S(\eta) + \frac{c\rho^{\eta-1}}{g(\eta+1)} \sum_{i=k}^{\infty} i\rho^i + \sum_{i=k}^{\infty} c\rho^{\eta+i} \\
&\le \frac{\rho^k}{1-\rho} \left(1 + c\rho^\eta + \frac{c\rho^{\eta-1}}{g(\eta+1)} \frac{k+1}{1-\rho} \right) \\
&\le \frac{\rho^k}{1-\rho} \left(1 + c + \frac{c(k+1)}{g(\eta+1) - f(\eta)} \right)
\end{aligned}$$

□

B.2 Missing proofs for *BilateralMatch(H)*

In the proofs of this section and of the next ones we will use the following facts: for any bounded, non-negative function $\xi : \mathbb{R}_+ \mapsto \mathbb{R}_+$ and any constant $u > 0$, in the limit where $p_H \to 0$,

Fact 1. *For $\eta = \frac{(\ln u) + \xi(p_H)}{p_H^2}$, we have $(1 - p_H^2)^\eta = \frac{e^{-\xi(p_H)}}{u} + O(p_H^2)$.*

Fact 2. *For any constants p_E, $r > 0$ and for $\eta = \frac{(\ln u) + \xi(p)}{p_H^2}$, we have $(1 - p_E p_H)^\eta = o(p_H^r)$.*

Fact 3. *For any constant p_E, and for $\eta = \frac{(\ln u) + \xi(p_H)}{p_E p_H}$, we have $(1 - p_H^2)^\eta = 1 - O(p_H)$.*

Fact 4. *For any constant p_E and for $\eta = \frac{(\ln u) + \xi(p)}{p_E p_H}$, we have $(1 - p_E p_H)^\eta = \frac{e^{-\xi(p)}}{u} + O(p_H)$.*

B.2.1 Proof of Theorem 3

Proof of Theorem 3. We first upper-bound $\mathbb{E}[H^{\mathcal{B}_H}]$. Let

$$v(\lambda_H, \lambda_E, p_E, p_H) = \begin{cases} \frac{1}{p_E p_H}\left(\ln\left(\frac{\lambda_E}{\lambda_E - \lambda_H}\right) + c_3\sqrt{p_H}\right) + p_H^{-3/4} & \text{when } \lambda_H < \lambda_E, \\ \frac{1}{p_H^2}\left(\ln\left(\frac{2\lambda_H}{\lambda_E + \lambda_H}\right) + c_4\sqrt{p_H}\right) + p_H^{-3/4} & \text{when } \lambda_H > \lambda_E. \end{cases}$$

Where c_3 and c_4 are the constants from Proposition 5. Using the equality $\mathbb{E}[H^{\mathcal{B}_H}] = \sum_{i=1}^{\infty} \mathbb{P}[H^{\mathcal{B}_H} \geq i]$, we have:

$$\begin{aligned} \mathbb{E}[H^{\mathcal{B}_H}] &= \sum_{i=1}^{v(\lambda_H,\lambda_E,p_E,p_H)-1} \mathbb{P}[H^{\mathcal{B}_H} \geq i] + \sum_{i=v(\lambda_H,\lambda_E,p_E,p_H)}^{\infty} \mathbb{P}[H^{\mathcal{B}_H} \geq i] \\ &\leq v(\lambda_H, \lambda_E, p_E, p_H) + \sum_{j=p_H^{-3/4}}^{\infty} \frac{\gamma(p_H)^j}{1-\gamma(p_H)} \\ &\leq v(\lambda_H, \lambda_E, p_E, p_H) + \frac{\gamma''(p_H)^{p_H^{-3/4}}}{(1-\gamma''(p_H))^2}. \end{aligned}$$

Where we denote $\gamma'' = \max(\gamma, \gamma')$ and we used the result from Proposition 5 : $\mathbb{P}[H^{\mathcal{B}_H} \geq v(\lambda_H, \lambda_E, p_E, p_H) + j] \leq \frac{\gamma''(p_H)^j}{1-\gamma''(p_H)}$.

Applying the fact that $\gamma''(p_H)^{p_H^{-3/4}} = \left(1 - \sqrt{p_H} + o(\sqrt{p_H})\right)^{p_H^{-3/4}} = o(p_H^2)$, and some algebra we get the following upper-bound on $\mathbb{E}[H^{\mathcal{B}_H}]$:

- If $\lambda_H < \lambda_E$, then $\mathbb{E}[H^{\mathcal{B}_H}] \leq \frac{\ln\left(\frac{\lambda_E}{\lambda_E - \lambda_H}\right)}{p_E p_H} + o\left(\frac{1}{p_H}\right)$.
- If $\lambda_H > \lambda_E$, then $\mathbb{E}[H^{\mathcal{B}_H}] \leq \frac{\ln\left(\frac{2\lambda_H}{\lambda_H + \lambda_E}\right)}{p_H^2} + o\left(\frac{1}{p_H^2}\right)$.

Now we proceed to lower-bound $\mathbb{E}[H^{\mathcal{B}_H}]$: Applying Markov inequality to $\mathbb{E}[H^{\mathcal{B}_H}]$ and using Proposition 4, we get the following lower-bound on $\mathbb{E}[H^{\mathcal{B}_H}]$:

- If $\lambda_H < \lambda_E$, then $\mathbb{E}[H^{\mathcal{B}_H}] \geq \frac{\ln\left(\frac{\lambda_E}{\lambda_E - \lambda_H}\right)}{p_E p_H} + o\left(\frac{1}{p_H}\right)$.
- If $\lambda_H > \lambda_E$, then $\mathbb{E}[H^{\mathcal{B}_H}] \geq \frac{\ln\left(\frac{2\lambda_H}{\lambda_H + \lambda_E}\right)}{p_H^2} + o\left(\frac{1}{p_H^2}\right)$.

This completes the proof. □

B.2.2 Proof of Corollaries 1 and 2

Proof of Corollary 1. Define $x = \lambda_H/\lambda_E$, and $f(x) = \frac{\ln\left(\frac{1}{1-x}\right)}{x}$. Note that the constant $\frac{\ln\left(\frac{\lambda_E}{\lambda_E-\lambda_H}\right)}{p_E\lambda_H} = \frac{f(x)}{\lambda_E p_E}$. It is easy check that $f'(x) > 0$ in $x \in (0,1)$, and therefore $f(x)$ is increasing in $x \in (0,1)$. □

Proof of Corollary 2. Define $y = \lambda_H/\lambda_E$, and $g(y) = \frac{\ln\left(\frac{2y}{1+y}\right)}{y}$. Note that the constant $\frac{\ln\left(\frac{2\lambda_H}{\lambda_H+\lambda_E}\right)}{\lambda_H} = \frac{g(y)}{\lambda_E}$. It is easy to check that $g'(y) > 0$ when $y \in (1, y^*)$, and $g'(y) < 0$ in $y > y^*$ where y^* is the solution of $g'(y) = 0$:

$$g'(y^*) = 0 \Leftrightarrow \frac{1}{y^*+1} = \ln\left(\frac{2y^*}{1+y^*}\right) \Leftrightarrow (y^*+1)\ln(2 - 2/(y^*+1)) = 1,$$

□

B.2.3 Proof of Proposition 5

Instead of proving Proposition 5, we prove a stronger result. This will be useful later on to prove an upper bound for E agents (see Lemma 13 in B.2.4).

Proposition 18. *propUpperHprioH Under $\mathcal{B}_H$, if $\lambda_H < \lambda_E$, for any non-negative bounded function $\xi(p_H)$, for all $k \geq 0$:*

$$\mathbb{P}\left[H^{\mathcal{B}_H} \geq \frac{1}{p_E p_H}\left(\ln\left(\frac{\lambda_E}{\lambda_E - \lambda_H}\right) + \xi(p_H)\right) + k\right] \leq \frac{\gamma(p_H)^k}{1-\gamma(p_H)},$$

where $\gamma(p_H) = \frac{\lambda_H}{\lambda_E - (\lambda_E - \lambda_H)e^{-\xi(p_H)}} + O(p_H)$.

If $\lambda_H > \lambda_E$, for any non-negative bounded function $\xi'(p_H)$, for all $k \geq 0$:

$$\mathbb{P}\left[H^{\mathcal{B}_H} \geq \frac{1}{p_H^2}\left(\ln\left(\frac{2\lambda_H}{\lambda_E + \lambda_H}\right) + \xi'(p_H)\right) + k\right] \leq \frac{\gamma'(p_H)^k}{1-\gamma'(p_H)},$$

where $\gamma'(p_H) = \frac{e^{-\xi'(p_H)}}{2-e^{-\xi'(p_H)}} + O(p_H^2)$.

Proof. We wish to apply Lemma 2 with $[X(t), Y(t)] = [H^{\mathcal{B}_H}(t), E^{\mathcal{B}_H}(t)]$, in the special case where $S(h) = \mathbb{N}$ for all h, and $c = \delta = 0$. This implies that we need to find functions $f(\cdot)$ and $g(\cdot)$ such that for all $e \in \mathbb{N}$, $f(h) \geq Q^{\mathcal{B}_H}([h,e],[h+1,e])$, and $g(h) \leq Q^{\mathcal{B}_H}([h,e],[h-1,e])$.

Let $f(h) = \lambda_H(1-p_H^2)^h$ and $g(h) = \lambda_H(1-(1-p_H^2)^h) + \lambda_E(1-(1-p_E p_H)^h)$. We have

$$\frac{f(h)}{g(h+1)} = \frac{\lambda_H(1-p_H^2)^h}{\lambda_H(1-(1-p_H^2)^{h+1}) + \lambda_E(1-(1-p_E p_H)^{h+1})}.$$

Case $\lambda_H < \lambda_E$: Take $\eta = \frac{1}{p_E p_H}\left(\ln\left(\frac{\lambda_E}{\lambda_E - \lambda_H}\right) + \xi(p_H)\right)$. Facts 3 and 4 imply respectively that $(1-p_H^2)^\eta = 1 + O(p_H)$ and $(1-p_E p_H)^\eta = \frac{\lambda_E - \lambda_H}{\lambda_E} e^{-\xi(p_H)} + O(p_H)$. This yields for all $h \geq \eta$:

$$\frac{f(h)}{g(h+1)} \leq \frac{\lambda_H}{\lambda_E - (\lambda_E - \lambda_H)e^{-\xi(p_H)}} + O(p_H) := \gamma(p_H).$$

Case $\lambda_H > \lambda_E$: Taking $\eta = \frac{1}{p_H^2}\left(\ln\left(\frac{2\lambda_H}{\lambda_E + \lambda_H}\right) + \xi'(p_H)\right)$. Facts 1 and 2 imply respectively $(1-p_H^2)^\eta = \frac{\lambda_E + \lambda_H}{2\lambda_H} e^{-\xi'(p_H)} + O(p_H^2)$ and $(1-p_E p_H)^\eta = o(p_H^2)$. This yields for all $h \geq \eta$:

$$\frac{f(h)}{g(h+1)} \leq \frac{\frac{1}{2}e^{-\xi'(p_H)}}{1 - \frac{1}{2}e^{-\xi'(p_H)}} + O(p_H^2) := \gamma'(p_H).$$

In both cases, we conclude by applying Lemma 2 with $\rho = \gamma(p_H)$ or $\rho = \gamma'(p_H)$.

□

Proof of Proposition 5. The proof of Proposition 5 is a consequence of Proposition 18:

- In the case where $\lambda_H < \lambda_E$, take $\xi(p_H) = \sqrt{p_H}\frac{\lambda_H}{\lambda_E - \lambda_H}$. This implies that $\gamma(p_H) = \frac{\lambda_H}{\lambda_E - (\lambda_E - \lambda_H)e^{-\xi(p_H)}} + O(p_H) = 1 - \sqrt{p_H} + O(\sqrt{p_H})$.
- In the case where $\lambda_H > \lambda_E$, take $\xi'(p_H) = 2\sqrt{p_H}$. This implies that $\gamma'(p_H) = \frac{e^{-\xi'(p_H)}}{2 - e^{-\xi'(p_H)}} + O(p_H^2) = 1 - \sqrt{p_H} + O(\sqrt{p_H})$.

□

B.2.4 Proof of Proposition 4

The proof of Proposition 4 requires a concentration bound on the number of E agents, which we state in Lemma 13.

Lemma 13. *Under $\mathcal{B}_H$, and assuming $p_H \leq p_E$, there exist constants C and $\zeta < 1$ (which only depend on λ_H, λ_E, and p_E) such that for any $k \geq 0$, there exists p such that for any $p_H < p$, we have:*

$$\mathbb{P}\left[E^{\mathcal{B}_H} \geq \frac{1}{\sqrt{p_H}} + k\right] \leq C\zeta^k.$$

Proof. The proof is based on a bound on the right-tail distribution of E agents in the market. To do this, we will apply Lemma 2 with $[X(t), Y(t)] = [E^{\mathcal{B}_H}(t), H^{\mathcal{B}_H}(t)]$. Therefore, we find an upper-bound $f(e) = \lambda_E(1-p_E^2)^e \geq Q^{\mathcal{B}_H}([h,e],[h,e+1])$ on the upward transition (5.3c). Similarly, we would like to find a lower-bound $g(e)$ on the downward transition (5.3d), but we cannot find one for any $h \in \mathbb{N}$. Therefore we need to restrict our attention to some subset $S(e) \subset \mathbb{N}$. Recall that $Q^{\mathcal{B}_H}([h,e],[h,e-1]) = \lambda_H(1-p_H^2)^h(1-(1-p_Ep_H)^e) + \lambda_E(1-p_Ep_H)^h(1-(1-p_E^2)^e))$.

Case $\lambda_H < \lambda_E$:

$$S(e) = \left\{h \in \mathbb{N} \mid h \leq \frac{1}{p_Ep_H}\left(\ln\left(\frac{\lambda_E}{\lambda_E - \lambda_H}\right) + \ln 2\right) + e\right\}.$$

Applying Proposition 18 with $\xi(p_H) = \ln 2$, we get that $\mathbb{P}[H^{\mathcal{B}_H} \notin S(e)] \leq \frac{\gamma(p_H)^e}{1-\gamma(p_H)} = c\delta^e$, where $c = \frac{\lambda_H+\lambda_E}{\lambda_E-\lambda_H}$, $\delta = \frac{2\lambda_H}{\lambda_E+\lambda_H}$. Fact 4 implies that for all $h \in S(e)$, $(1-p_Ep_H)^h \geq \frac{\lambda_E-\lambda_H}{2\lambda_E}(1-p_Ep_H)^e + O(p_H)$. Keeping only the second term in $Q^{\mathcal{B}_H}([h,e],[h,e-1])$, we get the lower-bound:

$$Q^{\mathcal{B}_H}([h,e],[h,e-1]) \geq \frac{\lambda_E - \lambda_H}{2}(1-p_Ep_H)^e(1-(1-p_E^2)^e) + O(p_H) := g(e)$$

This yields:

$$\frac{f(e)}{g(e+1)} = \frac{2\lambda_E}{(\lambda_E - \lambda_H)(1-(1-p_E^2)^{e+1})}\left(\frac{1-p_E^2}{1-p_Ep_H}\right)^{e+1} + O(p_H).$$

We get for all $e \geq \frac{1}{\sqrt{p_H}}$, and p_H small enough, $\frac{f(e)}{g(e+1)} \leq \delta$. Furthermore, for $\rho = \frac{1+\delta}{2}$ and p_H

small enough,

$$\frac{\delta^e}{g(e+1)} \leq \delta^e \frac{(1-p_Ep_H)^{\frac{1}{\sqrt{p_H}}-e-1}}{g(\frac{1}{\sqrt{p_H}}+1)} \leq \left(\frac{\delta}{1-p_Ep_H}\right)^e \frac{1}{g(\frac{1}{\sqrt{p_H}}+1)} \leq \frac{\rho^e}{g(\frac{1}{\sqrt{p_H}}+1)}$$

We can apply Lemma 2 which yields the desired bound:

$$\begin{aligned}\mathbb{P}\left[E^{\mathcal{B}_H} \geq \frac{1}{\sqrt{p_H}} + k\right] &\leq \frac{\rho^k}{1-\rho}\left(1 + c + \frac{c(k+1)}{g(\frac{1}{\sqrt{p_H}}+1) - f(\frac{1}{\sqrt{p_H}})}\right) \\ &\leq \frac{\rho^k}{1-\rho}\left(1 + c + \frac{c(k+1)(\lambda_H+\lambda_E)}{2\lambda_H g(1/\sqrt{p_H}+1)}\right). \\ &\leq \frac{\rho^k}{1-\rho}\left(1 + c + \frac{c(k+1)(\lambda_H+\lambda_E)}{\lambda_H(\lambda_E-\lambda_H)}\right).\end{aligned}$$

Where we used first the fact that $\frac{f(1/\sqrt{p_H})}{g(1/\sqrt{p_H}+1)} \leq \delta = \frac{2\lambda_H}{\lambda_E+\lambda_H}$ and therefore $g(1/\sqrt{p_H}+1) - f(1/\sqrt{p_H}) \geq \frac{\lambda_E-\lambda_H}{\lambda_E+\lambda_H}$ and second the fact that $g(1/\sqrt{p_H}+1) = \frac{\lambda_E-\lambda_H}{2} + O(p_H)$.

Case $\lambda_H > \lambda_E$:

$$S(e) = \left\{h \in \mathbb{N} \mid h \leq \frac{1}{p_H^2}\left(\ln\left(\frac{2\lambda_H}{\lambda_E+\lambda_H}\right) + \ln 2\right) + e\right\}.$$

Applying Proposition 18 with $\xi(p_H) = \ln(2)$, we have $\mathbb{P}[H^{\mathcal{B}_E} \notin S(e)] \leq \frac{\gamma(p_H)^e}{1-\gamma(p_H)} = c\delta^e$, with $c = 3/2$ and $\delta = 1/3$. Fact 1 implies that for all $h \in S(e)$, $(1-p_H^2)^h \geq \frac{\lambda_H+\lambda_E}{4\lambda_H}(1-p_H^2)^e + O(p_H^2)$. Keeping only the first term in $Q^{\mathcal{B}_H}([h,e],[h,e-1])$, we get the lower-bound:

$$Q^{\mathcal{B}_H}([h,e],[h,e-1]) \geq \frac{\lambda_H+\lambda_E}{4\lambda_H}(1-p_H^2)^e(1-(1-p_Ep_H)^e) + O(p_H^2) = g(e).$$

This yields:

$$\frac{f(e)}{g(e+1)} \leq \frac{4\lambda_H\lambda_E}{(\lambda_H+\lambda_E)(1-(1-p_Ep_H)^e)}\left(\frac{1-p_E^2}{1-p_H^2}\right)^e + O(p_H^2).$$

Furthermore, with $\rho = 1/2$ we get that for p_H small enough, $\frac{\delta}{1-p_H^2} \leq \rho$. This leads to

$$\frac{\delta^e}{g(e+1)} \leq \delta^e \frac{(1-p_H^2)^{p_H^{-1/2}-e}}{g\left(p_H^{-1/2}+1\right)} \leq \left(\frac{\delta}{1-p_H^2}\right)^e \frac{1}{g\left(p_H^{-1/2}+1\right)} \leq \frac{\rho^e}{g\left(p_H^{-1/2}+1\right)}.$$

Therefore we can apply Lemma 2:

$$\begin{aligned}\mathbb{P}\left[E^{\mathcal{B}_H} \geq n_E + k\right] &\leq \frac{\rho^k}{1-\rho}\left(1 + c + \frac{c(k+1)}{g\left(p_H^{-1/2}+1\right) - f(p_H^{-1/2})}\right)\\ &\leq \frac{\rho^k}{1-\rho}\left(1 + c + \frac{c(k+1)3}{2g(p_H^{-1/2})}\right)\\ &\leq \frac{\rho^k}{1-\rho}\left(1 + c + \frac{6c(k+1)}{\lambda_H + \lambda_E}\right)\end{aligned}$$

Where we used the fact that for all $e \geq \frac{1}{\sqrt{p_H}}$, $\frac{f(e)}{g(e+1)} = o(p_H)$. Therefore for p_H small enough, $\frac{f(e)}{g(e+1)} \leq \frac{1}{3}$ and therefore $g(e+1) - f(e) \geq \frac{2}{3}g(e+1)$. Which concludes the proof.

□

We can now prove Proposition 4. The main idea is to apply Lemma 1, in two cases separately, where in one case we have few ($\leq \frac{2}{\sqrt{p_H}}$) E agents, and in the other case where we have many. Using Lemma 13, we can exponentially bound the second case.

Proof of Proposition 4. We will apply Lemma 1. Consider $S = \left\{e \in \mathbb{R}, e \leq p_H^{-1/2} + p_H^{-1/2}\right\}$. Using Lemma 13 with $k = p_H^{-1/2}$, we get that $\mathbb{P}[E^{\mathcal{B}_H} \notin S] = o(p_H^4)$.

For $e \in S$, we have

$$(1 - p_E p_H)^e \geq 1 - p_E p_H \left(p_H^{-1/2} + p_H^{-1/2} + o(\sqrt{p_H})\right) = 1 - 2p_E\sqrt{p_H} + o(\sqrt{p_H}).$$

Taking $f(h) = \lambda_H(1-p_H^2)^h\left(1 - 2p_E\sqrt{p_H}\right) + o(\sqrt{p_H})$, we have for $e \in S$, $Q^{\mathcal{B}_H}([h,e],[h+1,e]) \geq f(h)$. Let $g(h) = Q^{\mathcal{B}_H}([h,e],[h-1,e]) = \lambda_H(1-(1-p_H^2)^h) + \lambda_E(1-(1-p_E p_H)^h)$, and note that $f(h)$ is non-increasing and $g(h)$ is non-decreasing. We have:

$$\frac{g(h+1)}{f(h)} = \frac{\lambda_H + \lambda_E - \lambda_H(1-p_H^2)^{h+1} - \lambda_E(1-p_Ep_H)^{h+1}}{\lambda_H(1-p_H^2)^h\left(1-2p_E\sqrt{p_H}+o(\sqrt{p_H})\right)}$$

In the case $\lambda_H < \lambda_E$, Take $\eta = \frac{1}{p_Ep_H}\left(\ln\left(\frac{\lambda_E}{\lambda_E-\lambda_H}\right) - c_1'\sqrt{p_H}\right)$. Using Facts 3 and 4, we have $(1-p_H^2)^\eta = 1+O(p_H)$ and $(1-p_Ep_H)^\eta = \frac{\lambda_E-\lambda_H}{\lambda_E}e^{c_1'\sqrt{p_H}}+O(p_H) = \frac{\lambda_E-\lambda_H}{\lambda_E}(1+c_1'\sqrt{p_H}+o(\sqrt{p_H}))$, therefore:

$$\begin{aligned}\frac{g(\eta+1)}{f(\eta)} &= \frac{\lambda_H + \lambda_E - \lambda_H - (\lambda_E-\lambda_H)(1+c_1'\sqrt{p_H}) + o(\sqrt{p_H})}{\lambda_H\left(1-2p_E\sqrt{p_H}\right)+O(p_H)}\\ &= 1 - \left(\frac{\lambda_E-\lambda_H}{\lambda_H}c_1' - 2p_E\right)\sqrt{p_H} + o(\sqrt{p_H}).\\ &= 1-\sqrt{p_H}+o(\sqrt{p_H}).\end{aligned}$$

Where we fixed $c_1' = \frac{\lambda_H(1+2p_E)}{\lambda_E-\lambda_H}$. Using Lemma 1 with $[X(t),Y(t)] = [H^{\mathcal{B}_H}(t), E^{\mathcal{B}_H}(t)]$, $k = p_H^{-3/4}$ and $\rho = \frac{g(\eta+1)}{f(\eta)}$, we get $\rho^k = o(p_H^2)$ and:

$$\begin{aligned}\mathbb{P}[H^{\mathcal{B}_H} \le \eta - k] &\le \eta\cdot o(p_H^4)\left(1+\frac{1}{f(\eta)-g(\eta+1)}\right) + \frac{\rho^k}{1-\rho}\\ &\le \eta\cdot o(p_H^4)\left(1+\frac{1}{f(\eta)\sqrt{p_H}+o(\sqrt{p_H})}\right) + \frac{o(p_H^2)}{\sqrt{p_H}+o(\sqrt{p_H})}\\ &= o(p_H).\end{aligned}$$

Taking $c_1 = c_1' + p_E$, this enables us to conclude that

$$\mathbb{P}[H^{\mathcal{B}_H} \le \frac{1}{p_Ep_H}\left(\ln\left(\frac{\lambda_E}{\lambda_E-\lambda_H}\right) - c_1p_H^{1/4}\right)] \le o(p_H).$$

In the case $\lambda_H > \lambda_E$, Let $\eta = \frac{1}{p_H^2}\left(\ln\left(\frac{2\lambda_H}{\lambda_H+\lambda_E}\right) - c_2'p_H^{1/4}\right)$. Using Facts 1 and 2, we have

$(1-p_H^2)^\eta = \frac{\lambda_H+\lambda_E}{2\lambda_H} e^{c_2' p_H^{1/4}} + O(p_H^2)$ and $(1-p_E p_H)^\eta = O(p_H^2)$. This implies that:

$$\begin{aligned}
\rho = \frac{g(\eta+1)}{f(\eta)} &= \frac{(\lambda_H+\lambda_E) - \frac{(\lambda_H+\lambda_E)}{2} e^{c_2'\sqrt{p_H}} + O(p_H^2)}{\frac{(\lambda_H+\lambda_E)}{2} e^{c_2'\sqrt{p_H}} \left(1-2p_E\sqrt{p_H}\right) + O(p_H^2)} \\
&= \frac{2-(1+c_2'\sqrt{p_H})}{(1+c_2'\sqrt{p_H})(1-2p_E\sqrt{p_H})} + o(\sqrt{p_H}) \\
&= 1-(2c_2'-2p_E)\sqrt{p_H} + o(\sqrt{p_H}) \\
&= 1-\sqrt{p_H} + o(\sqrt{p_H}).
\end{aligned}$$

Where we have chosen $c_2' = \frac{1+2p_E}{2}$. Taking $k = p_H^{-3/4}$, we have $\rho^k = o(p_H^2)$. Applying Lemma 1, we get that for all :

$$\begin{aligned}
\mathbb{P}[H^{\mathcal{B}_H} \leq \eta - k] &\leq \eta \cdot o(p_H^4)\left(1+\frac{1}{f(\eta)-g(\eta+1)}\right) + \frac{\rho^k}{1-\rho} \\
&= \eta \cdot o(p_H^3) + \frac{o(p_H^2)}{f(\eta)\sqrt{p_H} + o(\sqrt{p_H})} \\
&= o(p_H)
\end{aligned}$$

Taking $c_2 = c_2' + 1$, this enables us to conclude that

$$\mathbb{P}[H^{\mathcal{B}_H} \leq \frac{1}{p_H^2}\left(\ln\left(\frac{2\lambda_H}{\lambda_H+\lambda_E}\right) - c_2\sqrt{p_H}\right)] \leq o(p_H).$$

□

B.3 Missing proofs for *BilateralMatch(E)*

The proof of Theorem 4 relies on a concentration result on the left tail of the distribution of $H^{\mathcal{B}_E}$ (Proposition 19), and a coupling argument to upper-bound $\mathbb{E}[H^{\mathcal{B}_E}]$ (Proposition 20).

Proposition 19. [Lower Bound] *Under $\mathcal{B}_E$ in steady-state,*

- *If $\lambda_H < \lambda_E$, there exist a function $\gamma(p_H) = 1-\sqrt{p_H}+o(\sqrt{p_H})$ and a constant $c_5 \geq 0$*

such that for all $k \geq 0$:

$$\mathbb{P}\left[H^{\mathcal{B}_E} \leq \frac{1}{p_E p_H}\left(\ln\left(\frac{\lambda_E}{\lambda_E - \lambda_H}\right) - c_5 p_H^{1/4}\right) - k\right] \leq \frac{\gamma(p_H)^k}{1 - \gamma(p_H)}.$$

- *If $\lambda_H > \lambda_E$, there exist a function $\gamma'(p_H) = 1 - \sqrt{p_H} + o(\sqrt{p_H})$ and a constant $c_6 \geq 0$ such that for all $k \geq 0$:*

$$\mathbb{P}\left[H^{\mathcal{B}_E} \leq \frac{1}{p_H^2}\left(\ln\left(\frac{2\lambda_H}{\lambda_H + \lambda_E}\right) - c_6\sqrt{p_H}\right) - k\right] \leq \frac{\gamma'(p_H)^k}{1 - \gamma'(p_H)}.$$

Proposition 20. [Upper-bound] *Under $\mathcal{B}_E$ and in steady-state,*

- *If $\lambda_H < \lambda_E$, then $\mathbb{E}[H^{\mathcal{B}_E}] \leq \frac{\ln\left(\frac{2\lambda_E}{\lambda_E - \lambda_H}\right)}{p_E p_H} + o\left(\frac{1}{p_H}\right)$*
- *If $\lambda_H > \lambda_E$, then $\mathbb{E}[H^{\mathcal{B}_E}] \leq \frac{1}{p_H^2}\ln\left(\frac{2\lambda_H}{\lambda_H + \lambda_E}\right) + o\left(\frac{1}{p_H^2}\right)$.*

Proof of Theorem 4. We will first compute a lower bound for $\mathbb{E}[H^{\mathcal{B}_E}]$: Applying Markov inequality to $\mathbb{E}[H^{\mathcal{B}_E}]$ and using Proposition 19 with $k = p_H^{-3/4}$, we get the following lower-bound on $\mathbb{E}[H^{\mathcal{B}_E}]$:

- If $\lambda_H < \lambda_E$, then $\mathbb{E}[H^{\mathcal{B}_E}] \geq \frac{\ln\left(\frac{\lambda_E}{\lambda_E - \lambda_H}\right)}{p_E p_H} + o\left(\frac{1}{p_H}\right)$.
- If $\lambda_H > \lambda_E$, then $\mathbb{E}[H^{\mathcal{B}_E}] \geq \frac{\ln\left(\frac{2\lambda_H}{\lambda_H + \lambda_E}\right)}{p_H^2} + o\left(\frac{1}{p_H^2}\right)$.

Using Proposition 20, we can get the desired upper-bounds for $\mathbb{E}[H^{\mathcal{B}_E}]$. We can conclude using Little's law: $w_H = \frac{\mathbb{E}[H^{\mathcal{B}_E}]}{\lambda_H}$.

□

B.3.1 Proof of Proposition 19.

In order to prove Proposition 19, we first need a concentration result on $E^{\mathcal{B}_E}$ agents. This is stated in Lemma 14.

Lemma 14. *Under $\mathcal{B}_E$ in steady-state, for any $\alpha \in [0,1]$, let $n_E(\alpha) = \frac{\ln(\alpha/(1+\alpha(1-p_E^2))}{\ln(1-p_E^2))}$. For any $k \geq 0$, we have:*

$$\mathbb{P}\left[E^{\mathcal{B}_E} \geq n_E(\alpha) + k\right] \leq \frac{\alpha^k}{1-\alpha}.$$

Also, $(1-p_E^2)^{n_E(\alpha)} = \frac{\alpha}{1+\alpha(1-p_E^2)}$.

Proof. In the same way that upper-bounding H agents when they get priority is easy because waiting E agents can be ignored, upper-bounding the number of E agents is easy when they get the priority because H agents can be ignored. We can get the following bounds on the transition probabilities: $Q^{\mathcal{B}_E}([h,e],[h,e+1]) \leq \lambda_E(1-p_E^2)^e = f(e)$, and $Q^{\mathcal{B}_E}([h,e],[h,e-1]) \geq \lambda_E(1-(1-p_E^2)^e) = g(e)$ which leads to:

$$\frac{f(e)}{g(e+1)} = \frac{(1-p_E^2)^e}{1-(1-p_E^2)^{e+1}}.$$

Setting $\eta = n_E(\alpha)$ to be the solution to $\frac{f(n)}{g(n+1)} = \alpha$, and applying Lemma 2 with $S(e) = \mathbb{N}$ and $[X(t), Y(t)] = [E(t), H(t)]$, for all h, $c = 0$ and $\delta = 0$, we get the desired result. □

We can now prove Proposition 19.

Proof of Proposition 19. Notice that $Q^{\mathcal{B}_E}([h,e],[h+1,e]) = Q^{\mathcal{B}_H}([h,e],[h+1,e])$, therefore the function $f(h) = \lambda_H(1-p_H^2)^h\left(1-2p_E\sqrt{p_H}\right) + o(\sqrt{p_H})$ used in the proof of Proposition 4 remains a lower bound for $e \in S = \left\{e \leq 2p_H^{-1/2}\right\}$. Furthermore, $Q^{\mathcal{B}_E}([h,e],[h-1,e]) \leq Q^{\mathcal{B}_H}([h,e],[h-1,e])$ therefore the function $g(h) = \lambda_H(1-(1-p_H^2)^h) + \lambda_E(1-(1-p_Ep_H)^h)$ remains an upper bound. Therefore, the proof is exactly the same as the proof for the lower bounds for Proposition 4, except that instead of Lemma 13 we use Lemma 14 in the special case $n_E(\alpha) = p_H^{-1/2}$ with $S = \left\{e \in \mathbb{R}, e \leq 2p_H^{-1/2}\right\}$, and $k = p_H^{-1/2}$. We still have, $\mathbb{P}[E^{\mathcal{B}_H} \notin S] = o(p_H^4)$. □

B.3.2 Proof of Propostion 20

This proof is based on a coupling argument: instead of analysing the CTMC resulted from $\mathcal{B}_E$, we analyse the CTMC underlying another process that we call $\widetilde{\mathcal{B}_E}$. $\widetilde{\mathcal{B}_E}$ works similarly to $\mathcal{B}_E$ with one crucial difference: each time an E agent arrives which cannot be matched immediately, it changes type and becomes an H agent and then joins the market. First note that under $\widetilde{\mathcal{B}_E}$, no E agent joins the market, making its underlying CTMC a 1-dimensional Markov chain. Proving that the stochasitc process underlying $\widetilde{\mathcal{B}_E}$ is a positive recurrent CTMC, and therefore it reaches steady-state is similar to the proof of positive recurrence of $\mathcal{B}_E$ (See Claim 17) and therefore omitted.

Using a coupling argument we show that in steady-state number of H agents under $\mathcal{B}_E$ can be upper-bounded by the number of H agents under $\widetilde{\mathcal{B}_E}$. Our motivation behind defining $\widetilde{\mathcal{B}_E}$ is that in $\mathcal{B}_E$, unmatched E agents in the market are competing against H agents over whom they have priority. The idea is that by turning unmatched E agents into H ones, we expect to have more H agents waiting. Let $H^{\widetilde{\mathcal{B}_E}}$ be the random number of H agents under $\widetilde{\mathcal{B}_E}$ in steady-state. First in the next lemma we show that $\mathbb{E}[H^{\mathcal{B}_E}] \leq \mathbb{E}[H^{\widetilde{\mathcal{B}_E}}] + 1$, then we compute an upper-bound on $\mathbb{E}[H^{\widetilde{\mathcal{B}_E}}]$.

Lemma 15. $\mathbb{E}[H^{\mathcal{B}_E}] \leq \mathbb{E}[H^{\widetilde{\mathcal{B}_E}}] + 1$.

Proof. Consider two copies of the arrival process, one under the setting of $\mathcal{B}_E$ and one under $\widetilde{\mathcal{B}_E}$. Let $[H_k^{\mathcal{B}_E}, E_k^{\mathcal{B}_E}]$ and $H_k^{\widetilde{\mathcal{B}_E}}$ denote the embedded (discrete-time) Markov chain resulting from observing the two dynamic systems at arrival epochs. We will prove the following (stronger) result: there exists a coupling such that at any step k, $H_k^{\mathcal{B}_E} + E_k^{\mathcal{B}_E} \leq H_k^{\widetilde{\mathcal{B}_E}} + 1$.

We start by coupling all arrivals: with probability $\frac{\lambda_H}{\lambda_E+\lambda_H}$ $(\frac{\lambda_E}{\lambda_E+\lambda_H})$, arrivals to $\mathcal{B}_E$ and $\widetilde{\mathcal{B}_E}$ at k are both H (E) agents. Three cases can happen:

1. $H_k^{\mathcal{B}_E} + E_k^{\mathcal{B}_E} < H_k^{\widetilde{\mathcal{B}_E}}$; in this case, we let the two chains evolve independently.
2. $H_k^{\widetilde{\mathcal{B}_E}} \leq H_k^{\mathcal{B}_E} + E_k^{\mathcal{B}_E} \leq H_k^{\widetilde{\mathcal{B}_E}} + 1$. Let $h = H_k^{\widetilde{\mathcal{B}_E}}$. We consider two sub-cases:
   (a) The arrival at $k+1$ is an H agent. We couple the events that $\widetilde{\mathcal{B}_E}$ and $\mathcal{B}_E$ cannot find a match as follows: we draw two independent Bernoulli random variables $\mathbb{B}_1$, $\mathbb{B}_2$ with respective parameters of $(1-p_H^2)^h$ and $(1-p_H^2)^{H_k^{\mathcal{B}_E}}(1-p_Hp_E)^{E_k^{\mathcal{B}_E}}(1-p_H^2)^{-h}$. $\mathbb{B}_1 = 1$ corresponds to the event that $\widetilde{\mathcal{B}_E}$ cannot find a match; similarly $\mathbb{B}_1\mathbb{B}_2 = 1$ corresponds to the event that $\mathcal{B}_E$ cannot find a match.

(b) The arrival at $k+1$ is an E agent. Similarly we couple the events that $\widetilde{\mathcal{B}_E}$ and $\mathcal{B}_E$ cannot find a match as follows: we draw two independent Bernoulli random variables $\mathbb{B}_3$, $\mathbb{B}_4$ with respective parameters of

$$(1-p_E p_H)^h \qquad \text{and} \qquad (1-p_E p_H)^{H_k^{\mathcal{B}_E}}(1-p_E^2)^{E_k^{\mathcal{B}_E}}(1-p_E p_H)^{-h}.$$

$\mathbb{B}_3 = 1$ corresponds to the event that $\widetilde{\mathcal{B}_E}$ cannot find a match. Similarly $\mathbb{B}_3\mathbb{B}_4 = 1$ corresponds to the event that $\mathcal{B}_E$ cannot find a match.

3. $H_k^{\mathcal{B}_E} + E_k^{\mathcal{B}_E} > H_k^{\widetilde{\mathcal{B}_E}} + 1$; in this case, we let the two chains evolve independently.

Having the above coupling, we finish the proof by induction. The base case $k = 0$ is trivial: $H_0^{\mathcal{B}_E} + E_0^{\mathcal{B}_E} = H_0^{\widetilde{\mathcal{B}_E}} = 0$. Suppose $H_k^{\mathcal{B}_E} + E_k^{\mathcal{B}_E} \leq H_k^{\widetilde{\mathcal{B}_E}} + 1$, we show that $H_{k+1}^{\mathcal{B}_E} + E_{k+1}^{\mathcal{B}_E} \leq H_{k+1}^{\widetilde{\mathcal{B}_E}} + 1$: In Case (1), note that the number of agents waiting can increase or decrease by at most 1. For both cases, we have $H_{k+1}^{\mathcal{B}_E} + E_{k+1}^{\mathcal{B}_E} \leq H_{k+1}^{\widetilde{\mathcal{B}_E}} + 1$. In Case (2a), we have:

- If $\mathbb{B}_1 = 0$, then $H_{k+1}^{\widetilde{\mathcal{B}_E}} = H_k^{\widetilde{\mathcal{B}_E}} - 1$. Further if $\mathbb{B}_1 = 0$ then $\mathbb{B}_1\mathbb{B}_2 = 0$, and $H_{k+1}^{\mathcal{B}_E} + E_{k+1}^{\mathcal{B}_E} = H_k^{\mathcal{B}_E} + E_k^{\mathcal{B}_E} - 1$. Therefore $H_{k+1}^{\mathcal{B}_E} + E_{k+1}^{\mathcal{B}_E} \leq H_{k+1}^{\widetilde{\mathcal{B}_E}} + 1$
- If $\mathbb{B}_1 = 1$, then $H_{k+1}^{\widetilde{\mathcal{B}_E}} = H_k^{\widetilde{\mathcal{B}_E}} + 1$. Note that number of agents under $\mathcal{B}_E$ can increase by at most one, therefore $H_{k+1}^{\mathcal{B}_E} + E_{k+1}^{\mathcal{B}_E} \leq H_{k+1}^{\widetilde{\mathcal{B}_E}} + 1$.

Similarly in Case (2b), we have:

- If $\mathbb{B}_3 = 0$, then $H_{k+1}^{\widetilde{\mathcal{B}_E}} = H_k^{\widetilde{\mathcal{B}_E}} - 1$. Further if $\mathbb{B}_3 = 0$ then $\mathbb{B}_3\mathbb{B}_4 = 0$, and $H_{k+1}^{\mathcal{B}_E} + E_{k+1}^{\mathcal{B}_E} = H_k^{\mathcal{B}_E} + E_k^{\mathcal{B}_E} - 1$; therfore $H_{k+1}^{\mathcal{B}_E} + E_{k+1}^{\mathcal{B}_E} \leq H_{k+1}^{\widetilde{\mathcal{B}_E}} + 1$
- If $\mathbb{B}_3 = 1$, then $H_{k+1}^{\widetilde{\mathcal{B}_E}} = H_k^{\widetilde{\mathcal{B}_E}} + 1$. Note that number of agents under $\mathcal{B}_E$ can increase by at most one, therefore and $H_{k+1}^{\mathcal{B}_E} + E_{k+1}^{\mathcal{B}_E} \leq H_{k+1}^{\widetilde{\mathcal{B}_E}} + 1$.

Thus, in all possible cases $H_{k+1}^{\mathcal{B}_E} + E_{k+1}^{\mathcal{B}_E} \leq H_{k+1}^{\widetilde{\mathcal{B}_E}} + 1$ finishing the proof. Note that the above induction also implies that Case (3) never occurs.

□

Proof of Proposition 20. Observe that the CTMC $H_t^{\widetilde{\mathcal{B}_E}}$ has the following rate matrix:

$$\begin{aligned}
Q^{\widetilde{\mathcal{B}_E}}(h,h+1) &= \lambda_H(1-p_H^2)^h + \lambda_E(1-p_Ep_H)^h \\
Q^{\widetilde{\mathcal{B}_E}}(h,h-1) &= \lambda_H(1-(1-p_H^2)^h) + \lambda_E(1-(1-p_Ep_H)^h)
\end{aligned}$$

And let us define:

$$\rho(h) = \frac{Q^{\widetilde{\mathcal{B}_E}}(h,h+1)}{Q^{\widetilde{\mathcal{B}_E}}(h+1,h)} = \frac{\lambda_H(1-p_H^2)^h + \lambda_E(1-p_Ep_H)^h}{\lambda_H(1-(1-p_H^2)^{h+1}) + \lambda_E(1-(1-p_Ep_H)^{h+1})}.$$

Note that $\rho(h)$ is a decreasing function, and suppose that there exists $\eta \geq 0$ such that $\rho(\eta) < 1 - \sqrt{p_H}$ and let $\pi = \pi(h)_{h\geq 0}$ be the stationnary distribution of $H^{\widetilde{\mathcal{B}_E}}$. Then we have for all $i \geq 0$, $\pi(\eta+i) \leq \rho(\eta)^i \pi(\eta) \leq \rho(\eta)^i$ and for all $k \geq 0$, $\mathbb{P}[H^{\widetilde{\mathcal{B}_E}} \geq \eta + k] \leq \pi(\eta)\sum_{i\geq k}\rho(\eta)^i \leq \frac{\rho(\eta)^k}{1-\rho(\eta)}$. We then have

$$\begin{aligned}
\mathbb{E}[H^{\widetilde{\mathcal{B}_E}}] &= \sum_{h\geq 1}\mathbb{P}[H^{\widetilde{\mathcal{B}_E}} \geq h] \\
\mathbb{E}[H^{\widetilde{\mathcal{B}_E}}] &\leq (\eta+k) + \sum_{h\geq \eta+k}\mathbb{P}[H^{\widetilde{\mathcal{B}_E}} \geq h] \\
\mathbb{E}[H^{\widetilde{\mathcal{B}_E}}] &\leq (\eta+k) + \frac{\rho(\eta)^k}{(1-\rho(\eta))^2}
\end{aligned}$$

Let us consider the case when $\lambda_H < \lambda_E$. Let $\eta = \frac{\ln(1/u)+c\sqrt{p_H}}{p_Ep_H}$. For $h \geq \eta$, we have $(1-p_Ep_H)^h \leq u\left(1 - c\sqrt{p_H} + o(\sqrt{p_H})\right)$ and $(1-p_H^2)^h \leq 1 - \ln(1/u)p_H^2 + o(p_H^2)$. We have $\rho(\eta) = \frac{\lambda_H + \lambda_E u(1-c\sqrt{p_H})}{\lambda_E(1-u(1-c\sqrt{p_H}))} + o(\sqrt{p_H})$. Taking $u = \frac{\lambda_E - \lambda_H}{2\lambda_E}$ and $c = \frac{\lambda_E+\lambda_H}{2(\lambda_E-\lambda_H)}$, we get $\rho(\eta) = 1 - \sqrt{p_H} + o(\sqrt{p_H})$. Taking $k = p_H^{-3/4}$ we get:

$$\mathbb{E}[H^{\widetilde{\mathcal{B}_E}}] \leq \eta + o(1/p_H) = \frac{\ln(\frac{2\lambda_E}{\lambda_E-\lambda_H})}{p_Ep_H} + o(1/p_H), \quad \lambda_H < \lambda_E. \tag{B.1}$$

In the case when $\lambda_H > \lambda_E$, let $\eta = \frac{\ln(1/u)+c\sqrt{p_H}}{p_H^2}$. For $h \geq \eta$, we have $(1-p_Ep_H)^h = o(p_H)$ and $(1-p_H^2)^h \leq u(1-c\sqrt{p_H}) + o(\sqrt{p_H})$. Taking $u = \frac{\lambda_H+\lambda_E}{2\lambda_H}$ and $c = 1/2$, we get $\rho(\eta) =$

$1-\sqrt{p_H}+o(\sqrt{p_H})$. Taking $k=p_H^{-3/4}$ we get:

$$\mathbb{E}[H^{\widetilde{\mathcal{B}_E}}] \leq \eta + o(1/p_H^2) = \frac{\ln(\frac{2\lambda_H}{\lambda_E+\lambda_H})}{p_H^2} + o(1/p_H^2), \quad \lambda_H > \lambda_E. \tag{B.2}$$

Putting Lemma 15 and upper-bounds given in (B.1) and (B.2) together finishes the proof.

□

B.4 Missing proofs for *ChainMatch(d)*

Proof of Proposition 2. As pointed at the end of Subsection 5.5.4, when $p_E = 1$, an arriving E agent is matched immediately by a bridge agent, implying that $E_t^{\mathcal{C}(d)} = 0$ and $H_t^{\mathcal{C}(d)} = H_t^{\hat{\mathcal{C}}(d)}$; consequently Proposition 6 implies the limit stated in the Proposition 2.

Further note that fixing λ_H, it is straightforward to check that function $f(\lambda_E) = \frac{\ln\left(\frac{\lambda_H}{\lambda_E}+1\right)}{\lambda_H}$ is decreasing for $\lambda_E > 0$. Similarly fixing λ_E, it is straightforward to check that function $g(\lambda_H) = \frac{\ln\left(\frac{\lambda_H}{\lambda_E}+1\right)}{\lambda_H}$ is decreasing for $\lambda_H > 0$. □

Proof of Corollary 3. First we observe that $\ln\left(\frac{\lambda_H}{\lambda_E(1-(1-p_E)^d)}+1\right)$ is decreasing in d. Therefore the worst upper-bound on $w_H^{\mathcal{C}(d)}$ is for $d=1$. Next we show that:

$$\begin{aligned} \ln\left(\frac{\lambda_H}{\lambda_E p_E}+1\right)^{p_E} &\leq \ln\left(\frac{\lambda_H}{\lambda_E}+1\right) \\ &\leq \ln\left(\frac{\lambda_H}{\lambda_E-\lambda_H}+1\right), \end{aligned}$$

where the first inequality holds because function $f(x) := (\frac{\lambda_H/\lambda_E}{x}+1)^x$ is increasing in $x \in (0,1]$. □

Proof of Proposition 3. In steady-state, for any function $f(\cdot,\cdot)$, we have

$$\mathbb{E}[f(H_k,E_k) - f(H_{k+1},E_{k+1})] = 0,$$

For $f(h,e) = h + e$, using our notation: $\Lambda = \lambda_H(1-(1-p_H)^d) + \lambda_E(1-(1-p_E)^d)$, we get:

$$\begin{aligned}
&0 = (\lambda_H(1-p_H)^d + \lambda_E(1-p_E)^d)\mathbb{E}[H_k + E_k - H_{k+1} - E_{k+1} | H_k + E_k < H_{k+1} + E_{k+1}] \\
&+ \Lambda\mathbb{E}[H_k + E_k - H_{k+1} - E_{k+1} | H_k + E_k \geq H_{k+1} + E_{k+1}] \\
&= -(\lambda_H(1-p_H)^d + \lambda_E(1-p_E)^d) + \Lambda\mathbb{E}[L - 1 \mid L \geq 1].
\end{aligned} \tag{B.3}$$

This gives us:

$$\mathbb{E}[L \mid L \geq 1] = \frac{\lambda_H(1-p_H)^d + \lambda_E(1-p_E)^d}{(\lambda_H(1-(1-p_H)^d) + \lambda_E(1-(1-p_E)^d))} + 1 = \frac{\lambda_H + \lambda_E(1-p_E)^d}{\lambda_E(1-(1-p_E)^d)} + 1 + o(1).$$

□

B.5 Heuristic argument for *ChainMatch(d)*

In what follows we provide a heuristic analysis of the CTMC underlying the *ChainMatch(d)*. This heuristic adds intuition for the behavior of the policy and further establishes what is supposedly the right constant in the case in which our theory only generates an upper bound ($p_E < 1$).

We introduce an auxiliary 3-dimensional CTMC, in which a chain-segment is not formed instantaneously; instead a chain-segment can only advance at certain "tokens" (or times) that arrive according to a Poisson process with rate μ[1]. We denote this auxiliary CTMC by $\widetilde{\mathcal{C}}(d,\mu)$ and its states by $[H_t^{\widetilde{\mathcal{C}}(d,\mu)}, E_t^{\widetilde{\mathcal{C}}(d,\mu)}, U_t] \in \mathbb{N}^2 \times \{0,1\}$. The first two dimensions represent the number of H and E agents. The third dimension U_t indicates whether a chain-segment is being conducted. Initially $U_0 = 0$. Suppose at time t_1 the first agent i arrives that is matched by an altruistic agent. At this time U_{t_1} becomes 1 indicating that a chain-segment formation is in process. The policy waits x unit of time for a token to arrive to find an agent

[1] This process is independent from the Poisson processes guiding the arrivals of H and E agents.

who can be matched by i (note that x is exponentially distributed with rate μ)[2]. If such an agent does not exist, then chain-segment is stopped; and U_{t_1+x} becomes 0. Otherwise U_{t_1+x} remains 1 and the same process repeats. The transition rates of $\widetilde{\mathcal{C}}(d,\mu)$ are:

$$Q^{\widetilde{\mathcal{C}}(d,\mu)}([h,e,0],[h+1,e,0]) = \lambda_H(1-p_H)^d \tag{B.4a}$$

$$Q^{\widetilde{\mathcal{C}}(d,\mu)}([h,e,0],[h,e+1,0]) = \lambda_E(1-p_E)^d \tag{B.4b}$$

$$Q^{\widetilde{\mathcal{C}}(d,\mu)}([h,e,0],[h,e,1]) = \lambda_H(1-(1-p_H)^d) + \lambda_E(1-(1-p_E)^d) \tag{B.4c}$$

$$Q^{\widetilde{\mathcal{C}}(d,\mu)}([h,e,1],[h,e,0]) = \mu(1-p_H)^h(1-p_E)^e \tag{B.4d}$$

$$Q^{\widetilde{\mathcal{C}}(d,\mu)}([h,e,1],[h-1,e,1]) = \mu\left(1-(1-p_H)^h\right) \tag{B.4e}$$

$$Q^{\widetilde{\mathcal{C}}(d,\mu)}([h,e,1],[h,e-1,1]) = \mu(1-p_H)^h\left(1-(1-p_E)^e\right) \tag{B.4f}$$

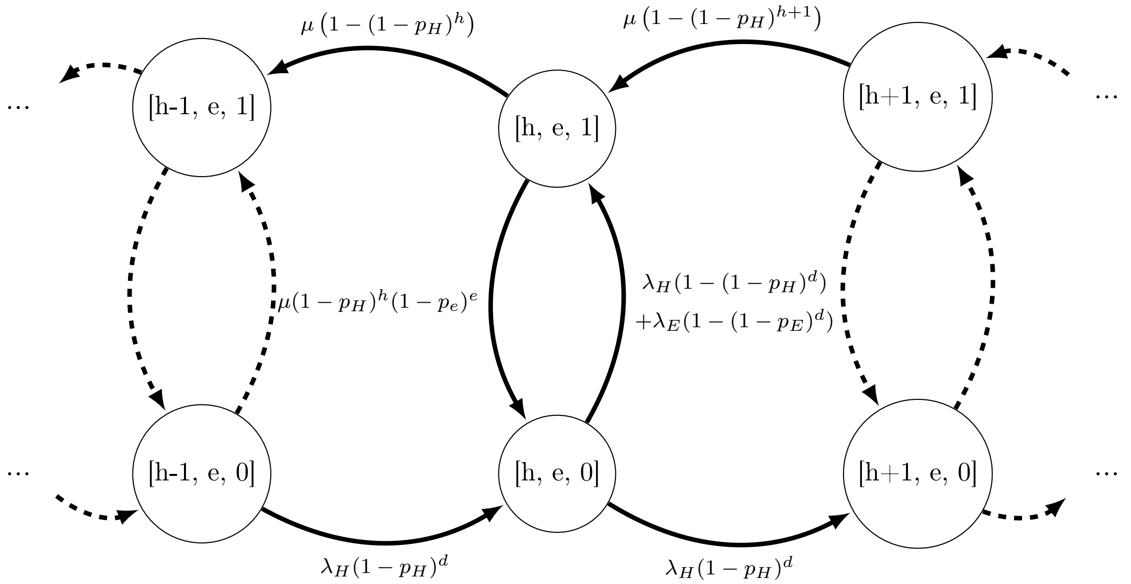

Figure B-1: Transitions for the CTMC underlying $\widetilde{\mathcal{C}}(d,\mu)$ in the first and third dimensions, i.e, transitions to states with a different number of E agents are not shown. Transition rates are given only for solid arrows.

[2] Note that it is possible that there is a new arrivals of H and/or E agents in the interval $(t_1, t_1+x]$. In this heuristic, we assume that the arriving H or E agent is deleted. The intuition is that when μ is large, this event happens rarely.

Figure B-1 illustrates the transitions of $\widetilde{\mathcal{C}}(d,\mu)$ in $H_t^{\widetilde{\mathcal{C}}(d,\mu)}$ and U_t dimensions along with the transition rates. The rate (B.4a) ((B.4b)) corresponds to the event that an H (E) agent arrives but cannot be matched by any bridge agent which happens with probability $(1-p_H)^d$ ($(1-p_E)^d$). The rate (B.4c) corresponds to the case an H arrives and starts a chain-segment (occurs with probability $(1-(1-p_H)^d)$) or an E agent arrives and starts a chain-segment (occurs with probability $(1-(1-p_E)^d)$; similarly (B.4d) represents the case where the chain-segment cannot advance any further (happens with probability $(1-p_H)^h(1-p_E)^e$). The last two rates correspond to adding one more H and E agent to the chain-segment respectively (with probability $\left(1-(1-p_H)^h\right)$ and $(1-p_H)^h\,(1-(1-p_E)^e)$ respectively).

Similar to the heuristic argument for *BilateralMatch(H)* and *BilateralMatch(E)*, we can try to solve the following system of nonlinear equations (that result from setting expected drifts in all three dimensions to be zero):

$$\begin{aligned}
&\mathbb{E}\left[\Lambda\mathbb{I}(U=0)-\mu(1-p_H)^{H^{\widetilde{\mathcal{C}}(d,\mu)}}(1-p_E)^{E^{\widetilde{\mathcal{C}}(d,\mu)}}\mathbb{I}(U=1)\right]=0\\
&\mathbb{E}\left[\lambda_E(1-p_E)^d\mathbb{I}(U=0)-\mu(1-p_H)^{H^{\widetilde{\mathcal{C}}(d,\mu)}}(1-(1-p_E)^{E^{\widetilde{\mathcal{C}}(d,\mu)}})\mathbb{I}(U=1)\right]=0\\
&\mathbb{E}\left[\lambda_H(1-p_H)^d\mathbb{I}(U=0)-\mu(1-(1-p_H)^{H^{\widetilde{\mathcal{C}}(d,\mu)}})\mathbb{I}(U=1)\right]=0,
\end{aligned}$$

where $\mathbb{I}(\cdot)$ is the indicator function. We find that $\ln\left(\frac{\lambda_H+\lambda_E}{\lambda_H(1-(1-p_H)^d)+\lambda_E}\right)/p_H$ is an approximate solution for $\mathbb{E}[H^{\widetilde{\mathcal{C}}(d,\mu)}]$. Further, note that in $\widetilde{\mathcal{C}}(d,\mu)$ if an H/E agent arrives while a chain-segment is being formed then the agent will not join the market. However, probability of such an event vanishes as $\mu\to\infty$ (i.e, forming chain-segments becomes instantaneous). Therefore, it is reasonable to approximate $\mathbb{E}[H^{\mathcal{C}(d)}]$ with $\lim_{\mu\to\infty}\mathbb{E}[H^{\widetilde{\mathcal{C}}(d,\mu)}]$, and thus with $\ln\left(\frac{\lambda_H+\lambda_E}{\lambda_H(1-(1-p_H)^d)+\lambda_E}\right)/p_H$. Finally note that:

$$\lim_{p_H\to 0}\ln\left(\frac{\lambda_H+\lambda_E}{\lambda_H(1-(1-p_H)^d)+\lambda_E}\right)=\ln\left(\frac{\lambda_H+\lambda_E}{\lambda_E}\right),$$

which is the constant in Proposition 2.

We emphasize that the analysis above is only a heuristic for guessing the right constant when $p_E < 1$. We refer the reader to Section 5.4.5 for numerical simulations that show the tightness of this constant.

B.6 Intuition and another heuristic for limit of $w_H^{\mathcal{C}(d)}$

In this section, we provide another heuristic argument that explains why p_E and d do not impact waiting time in the limit.

Let us look at the formation of a chain-segment starting with an E agent; it will have the following structure: $E - H - \ldots - H - E - H - \ldots - H - E - H - \ldots - H$; in other words, the chain-segment can be divided into sub-segments, each starting with an E agent, and following by a number of H agents until the sub-segment gets "stuck", meaning that there is no H agent that can continue the sub-segment (Recall that priority is given to H agents; therefore we will not search among E agents as long as there is an H agent that can continue the sub-segment). At that stage, the policy will look for an E agent to start a new sub-segment. Let Σ_E be the number of H agents in each sub-segment (between two E agents).

Next, let us look at the formation of a chain-segment starting with an H agent; the chain-segment has the following structure: $H - H - \ldots - H - E - H - \ldots - H - E - H - \ldots - H$, i.e, the first sub-segment starts with an H, but all the subsequent ones start with an E. Let us denote the expected number of H agents in the very first sub-segment by Σ_H.

With the above, let us consider the rate at which H agents join the market and the rate at which they depart. The former is simply $\lambda_H(1-p_H)^d$. The latter is $\lambda_E\mathbb{E}[\Sigma_E] + \lambda_H[1-(1-p_H)^d](\mathbb{E}[\Sigma_H]-1)$. Thus by conservation of H agents, we have:

$$(1-p_H)^d\lambda_H = \lambda_E\mathbb{E}[\Sigma_E] + \lambda_H[1-(1-p_H)^d](\mathbb{E}[\Sigma_H]-1).$$

Note that Proposition 3 - which shows that the expected length of a chain-segment is a constant - implies that $\mathbb{E}[\Sigma_H]$ is a constant, because $\mathbb{E}[\Sigma_H] \leq \mathbb{E}[L|L \geq 1]$, and Proposition 3 states that $\mathbb{E}[L|L \geq 1]$ remains a constant as $p_H \to 0$. Therefore, we can conclude that:

$$\mathbb{E}[\Sigma_E] = \lambda_H/\lambda_E + o(1),$$

i.e., in the limit $p_H \to 0$, almost all H agents match through sub-segments initiated by an E, and the length of such sub-segment only depends on the arrival rates. In other words, if we view the number of H agents that an E agent can help match as the "usefulness" of the E, this already suggests that (in the limit) the usefulness of an E does not depend on p_E nor on d.

However, we note that the time at which an E agent initiates a sub-segment changes with p_E:

- When $p_E = 1$, an arriving E agent initiates the first (and only) sub-segment of the chain-segment.
- When $p_E < 1$:
    - With probability $(1-(1-p_E)^d)$, the E agent is matched by the bridge agent, and initiates the 1-st sub-segment.
    - With probability $(1-p_E)^d$, the E agent joins the market, and initiates a sub-segment (that is not the first) at a later time.

Next, we analyze $\mathbb{E}[\Sigma_E]$ and relate it to the expected number of H agents. Recall that we denote the random number of H agents in steady-state by $H^{\mathcal{C}(d)}$. Consider the first sub-segment formed upon arrival of an E agent that receives from the bridge agent. We denote this sub-segment by Σ_1. Observe that

$$\mathbb{E}[\Sigma_1|H^{\mathcal{C}(d)} = h] = \sum_{i=1}^{h} \prod_{j=0}^{i-1} (1-(1-p_H)^{h-j}) := g(H^{\mathcal{C}(d)} = h). \tag{B.6}$$

Note that this equation is the direct consequence of giving priority to H agents. Therefore, $\mathbb{E}[\Sigma_1] = \mathbb{E}[\mathbb{E}[\Sigma_1 | H^{\mathcal{C}(d)} = h]] = \mathbb{E}[g(H^{\mathcal{C}(d)})]$. Note that $g(H^{\mathcal{C}(d)})$ does not depend on p_E or d. We make the following observations:

1. Function $g(\cdot)$ is a monotone increasing function, therefore it has a well-defined inverse function. Further, we have:

   Claim 15. *For $y > x$, we have $g(y) \geq g(x) + \ln\left(\frac{1}{1-p_H}\right)(y - x)$.*

   The above claim implies that if $g(y) - g(x) = o(1)$, then $y - x = o(1/p_H)$. The proof of the claim follows the same induction steps as that of Lemma 16; note that function $g(\cdot)$ has the same form as function $f(\cdot)$ in that lemma. Therefore, we omit the proof of the above claim.

2. For p_H small enough, $\mathbb{E}[g(H^{\mathcal{C}(d)})] = g(\mathbb{E}[H^{\mathcal{C}(d)}]) + o(1)$. Even though we cannot prove this concentration result, in Figure B-2, we provide numerical simulation verifying it for different p_E's.

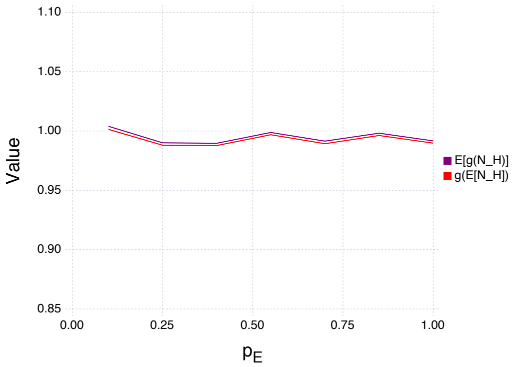

Figure B-2: Values of $g(\mathbb{E}[H^{\mathcal{C}(d)}])$ and $\mathbb{E}[g(H^{\mathcal{C}(d)})]$ as a function of p_E, for $p_H = 0.002$, $\lambda_H = 2$, $\lambda_E = 1$, $d = 1$ and $T = 10^6$.

Finally we claim that for p_H small enough, $\mathbb{E}[\Sigma_E] \approx \mathbb{E}[\Sigma_1]$; this follows from two observations:

1. $\mathbb{E}[\Sigma_1]$ is a constant; to see this, we use Proposition 2 which shows that the expected length of a chain-segment is a constant. Given that $\mathbb{E}[\Sigma_1] \leq \mathbb{E}[L|L \geq 1]$, we can conclude that $\mathbb{E}[\Sigma_1]$ is also a constant (L denotes the chain-segment length as it is defined above Proposition 2). Therefore when the second sub-segment is being formed, the number of H agents in the market will be $H^{\mathcal{C}(d)} - \Theta(1) \approx H^{\mathcal{C}(d)}$ (remember that $H^{\mathcal{C}(d)} = \Theta(1/p_H)$ following the upper-bound of Theorem 3 and the lower-bound of Appendix B.9).

2. The number of sub-segments is a constant: this follows from the observation that number of E agents in the market is an upper-bound on the number of sub-segments. In Lemma 16, we show that the expected number of E agents is a constant (independent of p_H).

Combining the above observations, for sufficiently small p_H, we have that:

$$g(\mathbb{E}[H^{\mathcal{C}(d)}(p_E = 1)]) + o(1) = \lambda_H/\lambda_E + o(1) = g(\mathbb{E}[H^{\mathcal{C}(d)}(p_E < 1)]) + o(1), \tag{B.7}$$

where with a slight abuse of notation we denoted the number of H agents in the market with parameter p_E by $H^{\mathcal{C}(d)}(p_E)$. Since $g(\cdot)$ has an inverse and using Claim 15, (B.7) implies that:

$$H^{\mathcal{C}(d)}(p_E = 1) = H^{\mathcal{C}(d)}(p_E < 1) + o(1/p_H). \tag{B.8}$$

We believe that the approximation becomes exact in the limit $p_H \to 0$. Even though the above argument is not rigorous, we hope that it sheds light on why p_E and d do not impact waiting time in the limit.

B.7 Positive recurrence proofs

To prove existence of a stationary distribution, we use a special case of a result from [87], as stated in [98].

Theorem 9 ([87])**.** *Suppose that X_t is an irreducible continuous time Markov chain, and suppose that there exist a nonnegative function V on S, a function $w \geq 1$ on S, a finite set $C \subset S$, and constants $c > 0$ and $b \in \mathbb{R}$ such that*

$$\sum_{j \in S} q_{ij} V(j) \leq -cw(i) + b \cdot \mathbb{I}_C(i) \text{ for all } i \in S,$$

where $\mathbb{I}_C$ denotes the indicator function of the set C. Then the Markov chain X is ergodic.

It is clear that all four markov chains are irreducible, so our proofs will focus on finding a suitable set C and function V for each case.

B.7.1 Positive recurrence of $\mathcal{B}_H$ and $\mathcal{B}_E$.

Claim 16. *The CTMC $[H_t^{\mathcal{B}_H}, E_t^{\mathcal{B}_H}]$ defined in* (5.3a)*,* (5.3b)*,* (5.3c)*,* (5.3d)*, is positive recurrent.*

Proof. Let $V([h,e]) = h + e$. Observe that for a continuous time random walk, we can write for any state $i = [h,e]$:

$$\begin{aligned}
\sum_{j \in \mathbb{N}^2} & q_{i,j} V(j) = \sum_{j \neq i} q_{i,j}(V(j) - V(i)) \\
&= Q([h,e],[h+1,e]) - Q([h,e],[h-1,e]) + Q([h,e],[h,e+1]) - Q([h,e],[h,e-1]). \\
&= \lambda_H (1-p_H^2)^h (1-p_E p_H)^e - \lambda_H (1-(1-p_H^2)^h) - \\
&\quad \lambda_E (1-(1-p_E p_H)^h) + \lambda_E (1-p_E p_H)^h (1-p_E^2)^e - \\
&\quad \lambda_H (1-p_H^2)^h (1-(1-p_E p_H)^e) - \lambda_E (1-p_E p_H)^h (1-(1-p_E^2)^e)) \\
&= 2\lambda_H (1-p_H^2)^h (1-p_E p_H)^e - \lambda_H + 2\lambda_E (1-p_E p_H)^h (1-p_E^2)^e - \lambda_E.
\end{aligned}$$

Take $M(p_H)$ such that $(1-p_H^2)^{M(p_H)} = \frac{1}{3}$, and let $C = \{[x,y] \text{ s.t. } x+y \leq 2M(p_H)\}$, note that C is finite. For any $[h,e] \notin C$, either $h \geq M(p_H)$ or $e \geq M(p_H)$. In both cases, we have $\sum_{j \in \mathbb{N}^2} q_{i,j} V(j) \leq -\frac{\lambda_H + \lambda_E}{3}$. □

Claim 17. *The CTMC* $[H_t^{\mathcal{B}_E}, E_t^{\mathcal{B}_E}]$ *defined in* (5.5a), (5.5b), (5.5c), (5.5d) *is irreducible and positive recurrent.*

The proof follows similar ideas as Claim 16.

B.7.2 Positive recurrence of $\mathcal{C}(d)$ and $\hat{\mathcal{C}}(d)$

Claim 18. *Under* $\mathcal{C}(d)$*, the number of* H *and* E *agents* $[H_t^{\mathcal{C}(d)}, E_t^{\mathcal{C}(d)}]$ *is a positive recurrent CTMC.*

Proof. Similarly to the proof of Claim 16, let $C = \{[x,y] \text{ s.t. } x+y \leq 2M\}$ for an appropriately chosen M and $V([h,e]) = h+e$. Consider a state $[h,e] \notin C$, and assume first that $h \geq M$. Denoting $\Lambda = \lambda_H(1-p_H)^d + \lambda_E(1-p_E)^d$, we have:

$$\begin{aligned}
\sum_{j \in \mathbb{N}^2} q_{i,j} V(j) &= \Lambda - (\lambda_H + \lambda_E - \Lambda) \sum_{[k_H,k_E] \leq [h,e]} (k_H + k_E) \mathbb{P}[k_H (k_E) \text{ H (E) agents get matched}] \\
&\leq \Lambda - (\lambda_H + \lambda_E - \Lambda) \sum_{[k_H] \leq [h,e]} k_H \mathbb{P}[k_H \text{ H agents get matched}] \\
&\leq \Lambda - (\lambda_H + \lambda_E - \Lambda) \sum_{k \leq h} \prod_{i=h}^{h-k} (1-(1-p_H)^i) \\
&\leq \Lambda - (\lambda_H + \lambda_E - \Lambda) \sum_{k \leq M/2} (1-(1-p_H)^{M/2})^k
\end{aligned}$$

Going from the second to the third inequality bounds the number of agents matched with the number of agents matched in the first sub-chain-segment (before matching an E agent). Because the function $M \to \sum_{k \leq M/2} (1-(1-p_H)^{M/2})^k$ tends to infinity as M grows large, we can find M such that $\sum_{j \in \mathbb{N}^2} q_{i,j} V(j) \leq -1$, which concludes the proof.

The case where $h < M$ and $e \geq M$ can be treated similarly using the following observation: every-time the chain cannot match to any H agent, it tries to match an E, and it succeeds with probability $1-(1-p_E)^e$ irrespective of h. Put together, these events constitute a set of E agents that are matched sequentially, and the total length of the chain is more than the number of E agents selected in that way. □

Claim 19. *$H_t^{\hat{C}(d)}$ is a positive recurrent CTMC.*

The proof follows similar arguments as that of Claim 18.

B.8 Lower bound

Definition 9. *We call a matching policy* anonymous Markovian *if matching decisions are made at arrival epochs, and only depend on the current compatibility graph $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$, and are anonymous among agents of the same type. In that case, the market $\mathcal{G}_t$ is a continuous-time Markov chain.*

We say that an anonymous Markovian policy $\mathcal{P}$ is stable *if the resulting Markov chain is ergodic and has a stationary distribution.*

In line with our previous notation, we denote $w_H^{\mathcal{P}}$ ($w_E^{\mathcal{P}}$) the expected stationary waiting times for H and E agents under policy $\mathcal{P}$.

Remark 5. *Observe that all the matching policies described in this paper are anonymous Markovian.*

Proposition 21. *For any $p_E \in [0,1]$, $\lambda_H > 0, \lambda_E \geq 0$, there exists a constant c such that for any $p_H > 0$, under any stable anonymous Markovian matching policy $\mathcal{P}$, $w_H^{\mathcal{P}} + w_E^{\mathcal{P}} \geq \frac{c}{p_H}$.*

The proof follows ideas used in [8]. The main intuition is the following: Suppose the market size is too small, then an arriving agent has to wait a *long* time to obtain at least one incoming edge. This long waiting time contradicts the small market size (with Little's law).

Proof. In this proof, we fix a Markovian policy $\mathcal{P}$, and we will omit the superscript notations. Observe that under $\mathcal{P}$, the market $\mathcal{G}_t$ only evolves at arrival epochs, and we can analyze the embedded discrete-time Markov chain resulting from observing the system at arrival epochs which we denote by $\mathcal{G}_i$. Let $n = \mathbb{E}[|\mathcal{V}|] = \mathbb{E}[H] + \mathbb{E}[E]$ be the expected size of the market in steady-state. Let us denote $\theta = \frac{\lambda_H}{\lambda_H + \lambda_E}$, and let $\hat{w}_H$ be the expected number of (discrete) time steps that an H agent arriving in steady-state has to wait before getting matched. Little's law for discrete Markov chains implies that $\hat{w}_H = \mathbb{E}[H]/\theta$.

Note that it is enough to prove that there exists a constant k such that $n \geq k/p_H$ (we then choose $c = \frac{k}{\max(\lambda_H, \lambda_E)}$). Let k be a constant to be defined later. Assume for contradiction that there exists p_H such that $n < k/p_H$. Let a be an H agent entering the market in steady-state. Let $\mathcal{V}_a$ be the set of agents in the market when agent a arrives. Note that we assumed $\mathbb{E}[|\mathcal{V}_a|] = n < k/p_H$. Define the event $E_1 = \{|\mathcal{V}_a| \leq 3n/\theta\}$. By Markov's inequality, $\mathbb{P}[E_1] \geq 1 - \frac{\mathbb{E}[|\mathcal{V}_a|]\theta}{3n} \geq 1 - \theta/3$.

Let $\mathcal{A}_a$ be the first $3n/\theta$ arrivals after a, and let E_2 be the event that at least one agent from $\mathcal{V}_a \cup \mathcal{A}_a$ has an outgoing edge towards a. We have

$$\mathbb{P}[E_2] = \mathbb{P}[\mathbf{Bin}(|\mathcal{V}_a| + |\mathcal{A}_a|, p_H) \geq 1].^3$$

Therefore we get:

$$\mathbb{P}[E_2|E_1] \leq \mathbb{P}[\mathbf{Bin}(6n/\theta, p_H) \geq 1] \leq \mathbb{P}[\mathrm{Bin}(6k/\theta p_H, p_H) \geq 1] \leq 6k/\theta.$$

Where the first inequality derives from the definition of E_1, the second uses the fact that $n \leq k/p_H$ and the third is Markov's inequality.

We now use the fact that if a does not have any edge from either $\mathcal{V}_a$ or $\mathcal{A}_a$, then she must wait longer than $3n/\theta$ time steps.

$$\hat{w}_H \geq \frac{3n}{\theta}\mathbb{P}[E_2^c] \geq \frac{3n}{\theta}\mathbb{P}[E_2^c|E_1]\mathbb{P}[E_1] \geq \frac{3n}{\theta}(1 - 6k/\theta)(1 - \theta/3) \geq \frac{3n}{\theta}(1 - 6k/\theta)(2/3).$$

[3] Note that we abuse the notation of **Bin(u,p)** by allowing its parameters to be random variables. In this case, conditional on the event $|\mathcal{V}_a| + |\mathcal{A}_a| = u$, the random variable $\mathbf{Bin}(|\mathcal{V}_a| + |\mathcal{A}_a|, p)$ has a binomial distribution with parameters u and p.

Thus we get:

$$n \geq \mathbb{E}[H] = \hat{w}_H \theta \geq 2n(1 - \frac{6k}{\theta}).$$

Therefore for $k = \frac{\theta}{24}$, we obtain a contradiction.

□

Observe that similar to [8], the above reasoning could be generalized to the case of *periodic* Markovian policies (see Definition 2 of [8]) which includes batching polices with constant batch size.

B.9 A lower bound on $w_H^{\mathcal{C}(d)}$

Here, we apply Proposition 21 to the *ChainMatch(d)* policy (which is a stable anonymous Markovian matching policy) to prove a lower bound on $w_H^{\mathcal{C}(d)}$:

$$w_H^{\mathcal{C}(d)} + w_E^{\mathcal{C}(d)} \geq \frac{c}{p_H}. \tag{B.9}$$

In the following lemma, we show that $w_E^{\mathcal{C}(d)} < k_E$, where k_E is a constant independent of p_H; therefore (B.9) implies that $w_H^{\mathcal{C}(d)} \geq \frac{c}{p_H} - k_E$, i.e., there exists a lower-bound on $w_H^{\mathcal{C}(d)}$ that scales with $1/p_H$.

Lemma 16. *Under ChainMatch(d) and for $0 < p_E \leq 1$, we have that $w_E^{\mathcal{C}(d)} < k_E$, where k_E is a constant independent of p_H.*

Proof. Let $[H^{\mathcal{C}(d)}, E^{\mathcal{C}(d)}]$ denote is the random number of H and E agents in steady-state. First observe that conditional on $E^{\mathcal{C}(d)} = x$, the expected number of E agents matched in a chain-segment is given by:

$$f(x) := \sum_{k=0}^{x} \prod_{i=0}^{k} (1 - (1 - p_E)^{x-i}).$$

Note that the above holds because under *ChainMatch(d)*, we give priority to H agents;

therefore, when computing the probability E agents in a chain-segment, we can ignore the H agents that are matched between two consecutive E agents.

Function $f(x)$ can be re-written recursively as $f(x) = (1-(1-p_E)^x)(1+f(x-1))$. We now wish to provide a lower bound of $f(x)$. Let α be a constant to be determined later, and let us assume by induction that $f(x-1) \geq \alpha(x-1)$. Then:

$$f(x) \geq (1-(1-p_E)^x)(1+\alpha(x-1)) = \alpha(x-1)+1-(1-p_E)^x(1+\alpha(x-1)).$$

Let $\xi_\alpha(y) = (1-p_E)^y(1+\alpha(y-1))$. Its derivative satisfies

$$\xi'_\alpha(y) = (1-p_E)^y \left(\ln(1-p_E)(1+\alpha(y-1))+\alpha\right).$$

Note that in $\xi_\alpha(y)$, y is a continuous variable. Let $\alpha = \ln(\frac{1}{1-p_E})$, for all $y \geq 1$, $\xi'_\alpha(y) \leq 0$. Further, $\xi_\alpha(1) \leq \xi_\alpha(0)$. Thus we have $\xi_\alpha(x) \leq \xi_\alpha(0)$, for $x \in \mathbb{N}$. Therefore we can complete our induction.

$$f(x) \geq \alpha(x-1)+1-\xi_\alpha(x) \geq \ln(\frac{1}{1-p_E})x.$$

We can now conclude our proof using conservation of E agents in the market:

$$\lambda_E(1-p_E)^d = (\lambda_E(1-(1-p_E)^d))+\lambda_H(1-(1-p_H)^d))\mathbb{E}[f(E^{\mathcal{C}(d)})] \geq \lambda_E(1-(1-p_E)^d)\ln(\frac{1}{1-p_E})\mathbb{E}[E^{\mathcal{C}(d)}]$$

where in the last inequality we use $f(x) \geq \ln(\frac{1}{1-p_E})x$ and $(1-(1-p_H)^d) \geq 0$. We take $k_E = \frac{(1-p_E)^d}{(1-(1-p_E)^d)\ln(\frac{1}{1-p_E})}$, which does not depend on p_H. □

B.10 Scaling of $w_H^{\mathcal{B}_H}$ and $w_H^{\mathcal{B}_E}$ when $\lambda_H = \lambda_E$

In Figure B-3, we present our numerical study on how $w_H^{\mathcal{B}_H}$ and $w_H^{\mathcal{B}_E}$ scale with p_H in the case that $\lambda_H = \lambda_E$; the figure plots $p_H w_H^{\mathcal{B}_H}$ and $p_H w_H^{\mathcal{B}_E}$ when p_H ranges form $10^{-3.5}$ to 10^{-7} while $p_E = 0.5$. As the plot shows, for p_H smaller than 10^{-5}, the normalized waiting time $p_H w_H^{\mathcal{B}_H}$ and $p_H w_H^{\mathcal{B}_E}$ both remain constant, which implies the asymptotic scaling is $1/p_H$ under both policies.

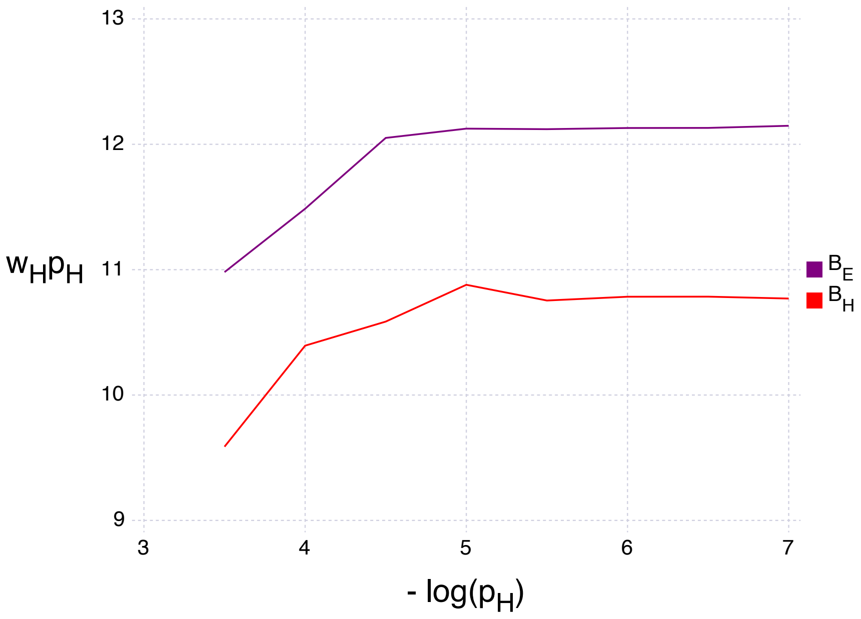

Figure B-3: Normalized waiting time $w_H p_H$ as a function of p_H (x axis is in log scale), for both $\mathcal{B}_H$, and $\mathcal{B}_E$. $T = 10^{12}$, $\lambda_H = \lambda_E = 1$, $p_E = 0.5$.

Appendix C

Appendix of Chapter 6

C.1 Proof of Claim 3

Proof of Claim 3. The proof of termination in [26] relies on the introduction of a minimum bid ϵ in step 6 of the auction algorithm to ensure that the algorithm does not get stuck in a cycle of bids of 0. In the limit where $\epsilon \to 0$, the algorithm ressembles the *hungarian algorithm* [74]. The idea is to search for an augmenting path along the edges for which the dual constraint is tight. If such a path is found, the matching is augmented, otherwise we perform simultaneous bid increases in way that ensures that prices p and margins q are still dual feasible.

We assume that we are given at time t an optimal matching m and optimal duals (p, q) corresponding to the graph with vertices S_t, B_t. We assume that we added a new vertex b^* to $B'_t = B_t \cup \{b^*\}$, and that we initialized $q_{b^*} = \max_{s \in S_t} v_{s,b^*} - p_s$

Initialize $m' = m$, $p' = p$, $q' = q$. Note that primal and dual feasibility are satisfied. Therefore, (m', p', q') is optimal iff the following three complementary slackness condition are satisfied:

$$\forall s \in S_t, v_{s,m(s)} = p'_s + q'_m(s). \tag{CS1}$$

$$\forall s \in S'_t, m(s) = \emptyset \implies p_s = 0. \tag{CS2}$$

$$\forall b \in B'_t, m(b) = \emptyset \implies q_b = 0. \tag{CS3}$$

Note that (CS1) and (CS2) are already satisfied. If $q'_{b^*} = 0$ then (CS3) is also satisfied and we have an optimal solution.

Suppose now that $q'_{b^*} > 0$. We will update (m', p', q') in a way that maintains primal and dual feasibility, as well as (CS1) and (CS2).

Our objective is to find an augmenting path in the graph. First we will start by trying to find an alternating path that starts on b and only uses edges for which the dual constraint is tight: $\mathcal{E} = \{(s, b) | s \in S'_t, b \in B'_t, v_{s,b} = p'_s + q'_b\}$. Observe that by (CS1) all the matched edges in m are in $\mathcal{E}$. We will now successively color vertices as follows:

0. Start by coloring b^* in blue.

1. For any blue buyer b, for any seller s such that $(s, b) \in \mathcal{E}$ and $s \neq m(b)$, we color s in red.

2. For any red seller s, let $b = m(s)$, then color b in blue.

Observe that there is an alternating path between b^* and any red seller. If at one point we color an unmatched seller s^* in red, this means that we have found an augmenting path from b^* to s^* that only utilizes edges in $\mathcal{E}$. In that case, we change m' according to the augmenting path. Because of the way we chose edges in $\mathcal{E}$, (CS1) is still satisfied. (CS2) and (CS3) are now also satisfied, which means we have an optimal solution (m', p', q').

We terminate when we are unable to color vertices any further. In that case, let us define $\delta_1 = \min_{b \text{ blue}} q_b$. If $\delta_1 = 0$, then there exists $b \in B'_t$ with $q_b = 0$ and an alternating path form b^* to b. We update m' according to that path, and verify that all CS conditions are now satisfied.

Suppose that $\delta_1 > 0$. Define

$$\delta_2 = \min_{b \text{ blue}, \, s \text{ not red}} \{p_s + q_b - v_{s,b}\}. \tag{C.1}$$

The fact that we cannot color any more vertices implies that $\delta_2 > 0$. Let $\delta = \min(\delta_1, \delta_2) >$

0. For every red seller s, we update $p'_s \leftarrow p'_s + \delta$. For every blue buyer b, we update $q'_b \leftarrow q'_b - \delta$. Observe that dual feasibility is still verified, as well as (CS1).

If $\delta = \delta_2$, taking (s, b) the argmin in (C.1), we now have such that $p'_s + q'_b - v_{s,b} = 0$ which means we can add (s, b) to $\mathcal{E}$ and color s in red. We will eventually have $\delta = \delta_1$, and this leads to $q_b = 0$ and we can terminate. This proves both the termination and correctness. Furthermore, monotonicity of the dual variables is also straightforward. Let us now prove the conservation property:

$$\sum_{s \in S_t} p_s + \sum_{b \in B_t} q_b = \sum_{s \in S_t} p'_s + \sum_{b \in B} q'_b. \tag{C.2}$$

Note that when we update the dual variables, then every seller we colored in red was matched in S' and we colored that match in blue. Therefore, apart from the initial vertex i, there are the same number of red and blue vertices. □

C.2 Proof of Claim 4

Proof of Claim 4. The idea of the proof is to iteratively apply the result of Claim 3 after any new arrival. Let $\widetilde{S}_t$ (resp. $\widetilde{B}_t$) be the set of sellers (buyers) who have departed, or already been matched before time t. We show by induction over $t \leq T$ that:

$$\sum_{s \in \widetilde{S}_t} p^f_s + \sum_{b \in \widetilde{B}_t} q^f_b + \sum_{s \in S_t} p_s(t) + \sum_{b \in B_t} q_b(t) = \sum_{b \in \widetilde{B}_t} q^i_b + \sum_{b \in B_t} q^i_b. \tag{C.3}$$

This is obvious for $t = 1$. Suppose that it is true for $t \in [1, T-1]$. Note that departures do not affect (C.3). If the agent arrivint at $t+1$ is a seller, then for all other sellers s, $p_s(t+1) = p_s(t)$ and for all buyers b, $q_b(t+1) = q_b(t)$, thus (C.3), is clearly still satisfied. Suppose that vertex $t+1$ is a buyer. Using equation (6.1), we have:

$$\sum_{s \in S_{t+1}} p_s(t+1) + \sum_{b \in B_{t+1}} q_b(t+1) = q_{t+1}(t+1) + \sum_{b \in B_t} q_b(t) + \sum_{s \in S_t} p_s(t) = \sum_{b \in B_{t+1}} q^i_b.$$

Note that at time $T + d$, every vertex has departed. Thus, $\widetilde{S}_{T+d} = S$, $\widetilde{B}_{T+d} = B$ and $S_{T+d} = B_{T+d} = \emptyset$. This enables us to conclude our induction and the proof for (C.3).

□

Appendix D

Appendix for Chapter 7

D.1 Proofs for random arrival order

Proof of Proposition 11. When n is not a multiple of p, let $v \in [1, p-1]$ be the remainder of the euclidian division of n by p, and u be such that $n = pu + v$. Let $\{\sigma_1, ..., \sigma_K\}$ be a p-periodic (α, d)-cover of $C^d_{n_1}$ with associated weights $\{\lambda_1, ..., \lambda_K\}$. We will show that it can be extended into an $(\alpha\,(u/u-2)\,, d)$-cover of C^d_n.

We set $\widetilde{\sigma}_k$ to be the p-periodic permutation over $1, ..., pu$ such that for all $i \in [1, p]$, $\widetilde{\sigma}_k(i) = \sigma_k(i)$. Let x be an integer in the interval $[1, u+1]$. Define the permutation $\sigma'_{k,x}$ as follows:

$$\sigma'_{k,x}(i) = \begin{cases} \widetilde{\sigma}_k(i) & i \leq px \\ i + (u-x)p & i \in [px+1, px+1+v] \\ \widetilde{\sigma}_k(i-v) & i > px+1+v. \end{cases} \tag{D.1}$$

Take $i', j' \in [1, px] \cup [px+1+v, n]$ such that $|i'-j'| \leq d$. Because $n_1 > p$ is a multiple of p, there exist $i, j \in [1, n_1]$ such that $i \equiv i' \mod p$, $j \equiv j' \mod p$ and $|i-j| \leq d$. By p-periodicity of σ_k and σ'_k, we know that edge (i', j') is in $B^d_n(\sigma'_k)$ iff (i, j) is in $B^d_{n_1}(\sigma_k)$. Thus we can conclude that $\sum_k \lambda_k B^d_n(\sigma'_{k,x})$ covers edge (i', j') of C^d_n.

Every edge is therefore covered for at least $u-2$ different values of x. Therefore, $\sum_k \sum_x \frac{u}{u-2} \lambda_k B^d_n(\sigma'_{k,x})$ covers C^d_n. This means that $(\sigma_{k,x})_{k,x}$ is an $(\alpha\,(u/u-2)\,, d)$-cover of C^d_n.

□

Proof of Proposition 12 in case (ii). Suppose now that $d+1 = ku+v$ with $1 \le v < k$. We first select vertices in the following way: select a subset $\Phi \subset [1, d+1]$ of $d+1-v$ vertices uniformly at random. Take $\Delta = \Phi + ku[1, r-1] = \{a + kub | a \in \Phi, b \in [1, r-1]\}$ and note that $|\Delta| = kur$.

We now contract vertices in Δ. This is the same as in Definition 8: for $t \in [1, u]$, a_t is the set of u smallest vertices of Δ that are not in $a_1 \cup ... \cup a_{t-1}$. Because $d+1-v$ is a multiple of u, we have $a_{i+k} = a_i + (d+1)$. This implies that the contracted graph is $C_{n/u}^{(d+1)/u}$.

Similarly to the proof of case (i), we extend a cover for $C_{n/u}^{(d+1)/u}$ to cover every edge (i, j) for $i, j \in \Delta$. If we sum over all the possible ways to select subset Φ, we note that every edge $(i, j) \in C_{r(d+1)}^{d}$ is covered with probability at least $\left(\frac{d+1-v}{d+1}\right)^2$. □

D.2 Random arrival order: numerical values for small d

Solving (LP$_d$) and (LP'$_k$) can become computationally difficult given the increase in the number of constraints, which is exponential in d. In Table D.1, we show numerical values of solutions α_d and α'_d for d between 2 and 13.

d	1	2	3	4	5	6	7	8	9	10	11	12	13
α_d	2	2.33	2.5	2.64	2.71	2.75	2.79	2.83	2.99^*	3.2^*	3.11^*		3.23^*
α'_d		4	3.45	3.17	3.15	3.12	3.09	3.08	3.07	3.20^*	3.15^*	3.26^*	3.32^*

Table D.1: Numerical values for α_d and α'_d for small values of d. Starred elements were solved approximately (are therefore upper bounds on the actual value).

We now need to provide upper bounds for all d between 14 and 51. Note first that if $d+1$ is a multiple of k, then Proposition 12 implies that $\alpha_d \le \alpha'_k$. Therefore, we need only consider prime values for d. In table D.2, we compute upper bounds for α_d using Proposition 12. For each case, we report which value of k we used, as well as the value $\alpha'_k \left({}^{(d+1-v)}/_{(d+1)}\right)^2$.

This allows us to conclude that Batching is 0.279-competitive.

d	17	19	23	29	31	37	41	43	47
$\alpha_d \leq$	3.58	3.48	3.44	3.31	3.36	3.30	3.31	3.24	3.35
k used	4	6	11	7	5	6	5	7	9

Table D.2: Upper bounds for α_d for prime values of d; derived from Proposition 12 using the following formula: $\alpha_d \leq \alpha_k \left(\frac{d+1-v}{d+1}\right)^2$ for $k \leq d$ and $v = d \bmod k$.

Appendix E

Appendix for Chapter 8

E.1 Upper bounds.

E.1.1 Upper bound for edge-weighted graphs.

Conjecture 4. *No algorithm is more than* $2/3$*-competitive on edge-weighted graphs.*

Proof. Consider the following *type graph* T with three vertices $\{1, 2, 3\}$ and two edges with weights $\{w_{1,2} = 1/\epsilon, w_{1,3} = 1\}$. Assume arrival rates are respectively $1/2$, ϵ and 1 and departures of $1/2$, 1 and 0 for vertices $1, 2$ and 3 (see Figure E-1).

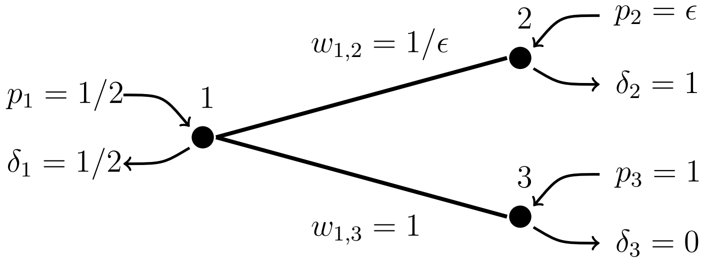

Figure E-1: Graph T with three vertices $\{1, 2, 3\}$ and two edges with weights $\{w_{1,2} = 1/\epsilon, w_{1,3} = 1\}$.

The proof is based on two claims.

Claim 20. *In the limit* $\epsilon \to 0$, $\mathcal{O} \geq 1.5 + o(1)$.

Claim 21 (Unproven)**.** *In the limit $\epsilon \to 0$, for any markovian algorithm $\mathcal{A}$, $\mathcal{A} \leq 1 + o(1)$*

□

Let us first show that the offline algorithm $\mathcal{O}$ obtains on average at least $1.5 + o(1)$ per time step (as $\epsilon \to 0$).

Proof of Claim 20. Let t be the time of arrival of a node of type 2. Let R be the (random) time since the previous arrival of a vertex of type 2. Note that $\mathbb{P}\left[R > k\right] = (1-\epsilon)^k$. Therefore $\mathbb{P}\left[R > \frac{1}{\sqrt{\epsilon}}\right] \geq 1 - \sqrt{\epsilon}$. Let X be the event that there has been an arrival of type 1 between $t - R$ and t that has not yet departed.

$$\mathbb{P}\left[X\right] \geq \sum_{k=0}^{R} \mathbb{P}\left[\text{ arrival at t+k }\right] \mathbb{P}\left[\text{ the arrival did not depart by t+R }\right] = \sum_{k=0}^{R} \frac{1}{2}\frac{1}{2^{R-k}} = 1 - \frac{1}{2^R}$$

Therefore $\mathbb{P}\left[X\right] \geq (1 - \sqrt{\epsilon})(1 - \frac{1}{2^{\sqrt{1/\epsilon}}}) = 1 + o(1)$ The OFF algorithm therefore keeps the node of type 1 that will stay until t (which yields 1 in expectation), and matches the others to 3 (yielding $1/2 - \epsilon$). Therefore $\mathcal{O} \geq 1.5 + o(1)$. □

Let us now show that any online algorithm $\mathcal{A} \leq 1 + o(1)$.

Proof Sketch of Claim 21. Observe that the state of the system only depends on the number of vertices of type 1. When a vertex of type 2 arrives, if there is a vertex of type 1 then it is a dominant strategy to match them. We can reduce the problem to that of finding the threshold M for the number N_1 of vertices of type 1 above which $\mathcal{A}$ decides to match an incoming vertex of type 1 to a vertex of type 3. Thus, based on the threshold M, any algorithm is between the greedy algorithm (which always matches 1 to 3) and the patient algorithm which always makes 1 wait.

□

E.2 Special case: star graphs

Consider a star graph with $n+1$ nodes. We assume w.l.o.g that $w_{0,1} \geq w_{0,2} \geq ... \geq w_{0,n-1}$. Figure E-2 illustrates the case $n = 5$.

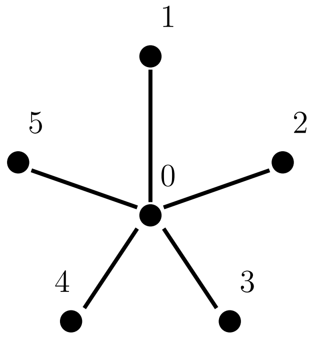

Figure E-2: Graph $G([0,5],\{(0,1),(0,2),(0,3),(0,4),(0,5)\})$.

The idea is to remove low-weight edges in way that ensures that the probability of the center node being matched is equal to $\frac{1}{2}$.

Let us start with the case where w.h.p. there is only one agent at the center at any time (this corresponds to a high departure to arrival ratio, which is precisely the case not handled in section 8.2.3)

Let $v_0 = p_0/2 + \delta_0 - p_0\delta_0/2$. [1] Consider i such that $p_1 + p_2 + ... + p_{i-1} \leq v_0 \leq p_1 + p_2 + ... + p_i$. Let us split node i into two new nodes i' and i'' such that $w_{0,i'} = w_{0,i''} = w_{0,i}$, $\delta_{i'} = \delta_{i''} = \delta_i$, and $p_{i'} = v_0 - (p_1 + p_2 + ... + p_{i-1})$, $p_{i''} = p_i - p_{i'}$. Note that this does not modify the performance of the offline algorithm.

We can now define our algorithm:

Proposition 22. *$\mathcal{A}$ yields a competitive ratio of $\frac{1-1/e}{16}$ on any star graph.*

Proof. Observe that for any node of type a_0 that decides to wait, the probability that it gets matched at a given time step is the same as the probability of departing/being discarded. Therefore, the probability that it is eventually matched is 1/2.

[1] v_0 represents the probability that either a node of type 0 arrives and decides to wait, or the current node of type 0 departs

Algorithm 8 $\mathcal{A}$

When a node of type $1, ..., i'$ arrives, match it to an node of type 0 if available, otherwise wait.
When a node of type $i'', i+1, ..., n$ arrives, make it wait.
When a node of type 0 arrives,

- With probability 1/2, wait and discard any previously arrived 0 nodes.
- With probability 1/2, match to the best available node, if none is available, discard it.

For a given instance, let us consider the set S_1 of nodes of type 0 that were matched immediately upon arrival by the Offline algorithm. Similarly, let S_2 be the set of nodes of type 0 that were matched after waiting at least one time step. For $x \in S_1 \cup S_2$, define m_x the node x was matched to, and $v_x = w_{0,m_x}$ the value of the match.[2] The value obtained by the offline algorithm is given by $OFF = \sum_{x \in S_1} v_x + \sum_{x \in S_2} v_x$.

E.2.1 Case 1: S_1

Consider a node $x \in S_1$. With probability 1/2, $\mathcal{A}$ also decided to match x immediately. It is possible that m_x has already been matched by $\mathcal{A}$, but we assume for now that $\mathcal{A}$ is allowed to double match m_x. Therefore w.p. 1/2, $\mathcal{A}$ obtains at least v_x.

E.2.2 Case 2: S_2

Definition 10. *We say that two nodes x and x' of type 0 are* adjacent *if there are present in the pool at the same time under the no-match policy. For a node x, $S(x)$ is the smallest set that contains x and such that for all $x' \in S(x)$, $S(x)$ contains all the nodes that are adjacent to x'.*

Lemma 17. *For all x, $S(x)$ is finite almost surely.*

Proof. The Markov chains that counts the number of nodes of type 0 under the no-match policy hits the value 0 infinitely often. □

[2] notation overload

Consider now a node $x \in S_2$.

With probability $1/2$, $\mathcal{A}$ also makes x wait. With probability greater than $1/2$, x is not yet matched when y arrives. If $k \leq i$, we obtain that $\mathcal{A}(x) = \frac{w_{a_0,a_k}}{2} = \frac{\mathcal{OFF}(x)}{2}$.

Observe that the number of matches to $k > i$ cannot be more than half of all nodes of type a_0. □

E.2.3 No match policy

For each $i \in [1, I]$ and a given matching policy, we further define N_i^t the number of nodes of type i in the system at time t.

Let us define $\mathcal{SS}$ to be the policy that never matches any node. Under $\mathcal{SS}$, each of the $N_i(\mathcal{SS}) = (N_i^t)_{t\geq 0}$ is an independent positive recurrent Markov chain. Let $\pi_i(\mathcal{SS}) = (\pi_i^k)_{k\geq 0}$ be the steady-state distribution of N_i.

Claim 22. $\pi_i^0(\mathcal{SS}) \leq \frac{\delta_i}{p_i+\delta_i}$.

Proof. Note that $\mathbb{P}[N_i(t+1) = 0|N_i^t = k] = \delta_i^k$. Therefore flow conservation yields:

$$\pi_i^0 p_i = \sum_{k\geq 1} \pi_i^k \delta_i^k \leq \sum_{k\geq 1} \pi_i^k \delta_i = (1 - \pi_i^0)\delta_i.$$

□

Claim 23. *If* $\delta_i \geq p_i$ *then* $\pi_i^0(\mathcal{SS}) \geq \frac{\delta_i - p_i}{\delta_i}$.

Proof. Note that $\mathbb{P}[N_i(t+1) \geq k+1|N_i^t \leq k] \leq p_i$, and that $\mathbb{P}[N_i(t+1) \leq k|N_i^t = k+1] \geq \delta_i$. Therefore flow conservation yields $\pi_i^k p_i \geq \pi_i^{k+1}\delta_i$, i.e. $\pi_i^{k+1} \leq \frac{p_i}{\delta_i}\pi_i^k$.

Thus, we have,

$$1 = \sum_{k\geq 0} \pi_i^k = \pi_i^0 + \sum_{k\geq 1} \pi_i^k \leq \pi_i^0 + \sum_{k\geq 1} \frac{p_i}{\delta_i}\pi_i^{k-1} = \pi_i^0 + \frac{p_i}{\delta_i}.$$

□