Batching and Greedy Policies: How Good Are They in Dynamic Matching?

Myungeun Eom Alejandro Toriello
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, Georgia, USA 30332

August 11, 2023

Abstract

We study a dynamic non-bipartite stochastic matching problem, where nodes appear following a type-specific independent distribution and wait in the system for a given sojourn time. This problem is motivated by applications in ride-sharing and freight transportation marketplaces, and is related to other on-demand marketplaces. We study the asymptotic properties of two widely used policies, batching and greedy, by analyzing a single-pair case and then converting to the general counterpart using a fluid relaxation and randomization. Finally, we present a computational study simulating freight transportation and ride-sharing marketplaces to assess the empirical effectiveness of the policies. We show that the batching policy is asymptotically optimal with respect to the sojourn time; similarly, while a straightforward greedy policy may not be optimal, a greedy policy with randomized modifications is asymptotically optimal. Perhaps more practically relevant, both policies converge exponentially fast to approximate optimality. We also extend our model to an impatient setting in which each unmatched node leaves at the end of each period with a type-dependent probability. We show that the results for the two policies still hold under different assumptions about the nodes' patience; roughly speaking, the batching policy requires more patient nodes than the greedy policy to remain optimal. Our results suggest that managers can achieve near-optimal performance by using simple greedy or batching policies, with only a reasonably small maximum waiting time guarantee, and even in the presence of potentially impatient nodes.

1 Introduction

Dynamic matching models arise in many applications, including organ exchanges (Anderson et al. 2017), e-commerce platforms and online advertisement (Blanchet et al. 2022). In transportation systems, dynamic matching models underpin applications such as ride-hailing (Zhang and Nie 2021), ride-sharing (Wang et al. 2018), and freight transportation marketplaces (Montecinos et al. 2021), where drivers and riders or multiple transport tasks are matched. Motivated by these applications, particularly ride-sharing and freight transportation markets, we study a dynamic non-bipartite stochastic matching problem.

The rise of the sharing economy has increased interest in dynamic matching systems, particularly in transportation platforms. As Feng et al. (2021) point out, system managers and policy makers struggle to understand the efficiency of on-demand transportation systems and its impact on agent waiting times. One way to assess the trade-off between increasing market thickness and agent delays is to fix a *sojourn period*, the maximum time an agent waits in the system, and optimize the system's average reward given this maximum. A fixed sojourn period enables the system manager to provide a service guarantee: agents will not wait longer than this time. Our main research question is whether the system can achieve near-optimal performance with a reasonably small sojourn period in dynamic matching markets.

Matching policies are another crucial factor that managers need to determine. To solve dynamic non-bipartite matching problems, several heuristics have been proposed in the literature and used in practice; batching and greedy policies are two of the most widely used (e.g. Yan et al. 2020). Roughly speaking, a batching policy accumulates a batch of arriving nodes within a fixed period of time and optimizes matches over each such batch. Conversely, a greedy policy makes matching decisions as opportunities arise by optimizing over available nodes. These policies are appealing because they are easy to implement, work well in practice, and do not require much detailed information or assumptions, such as knowledge of node arrival distributions. In spite of their prevalence, there have only been a limited number of studies analyzing the performance of these policies (e.g. Anderson et al. 2017, Aouad and Saritaç 2022, Ashlagi et al. 2022). In this work, we analyze the asymptotic properties of both.

Consider a freight transportation market, where shipment requests dynamically appear over time and a broker platform either assigns or auctions these requests to carriers. Since carriers (usually truck drivers) want to avoid *deadheading*, i.e. driving without a paying load, the platform may endeavor to bundle requests to reduce the deadhead distance, which reduces the platform's overall shipping costs. In principle, any number of requests can be bundled, but grouping more than two requests may introduce operational complexities, such as the potential for cascading effects if the first shipment in a bundle is delayed; for these and other reasons, the platform may prefer to bundle only pairs of requests.

We can model the platform's system as a dynamic non-bipartite matching problem in which nodes are shipment requests and the matching reward for a request pair is proportional to the deadhead distance saved if the same vehicle performs the requests sequentially instead of having each request served by a different vehicle that must deadhead back to its origin. An analogous model applies to ride-sharing and other similar applications, using different matching rewards. We study the case in which node arrivals in each period follow a type-specific, i.i.d. distribution and wait in the system for a given uniform sojourn period. Our concrete motivation comes from transportation marketplace applications, where the sojourn time represents the maximum waiting time the platform guarantees to shippers before a load is transported.

1.1 Contributions

Assuming the objective is to maximize the long-run average reward, our main contribution is to study the asymptotic properties of batching and greedy policies in a setting where node arrivals follow type-specific independent distributions and each node remains in the system for a fixed sojourn period $\tau \in \mathbb{N}$. The **batching policy** solves a maximum-reward matching problem and clears out the system every τ periods. The **greedy policy** solves a maximum-reward matching problem every period given the set of available nodes. A straightforward implementation of the greedy policy may not be optimal, so in our analysis we randomly restrict each node's compatibility based on a fluid relaxation of the system. To simplify terminology, we refer to our randomized greedy policy simply as a greedy policy when there is no danger of confusion.

We preview the main results in Table 1; we denote the average reward of the batching, greedy, and optimal policies respectively as B, G, and OPT. We note that the batching policy is oblivious to node type arrival distributions and uses only rewards and the sojourn time; the randomized greedy policy also requires expected arrivals per period by type, but no other distributional information.

Policy	Result	Arrival Dist.
Batching	$B(\tau) \geq OPT(\tau) - \mathscr{O}(1/\sqrt{\tau})$	finite variance
Batching	$B(\tau) \geq (1-\varepsilon)OPT(\tau) - \mathscr{O}(e^{-\tau\varepsilon})$ for small $\varepsilon > 0$	bounded support
Randomized Greedy	$G(\tau) \geq OPT(\tau) - \mathscr{O}(1/\tau)$	bounded support
Randomized Greedy	$G(\tau) \geq (1-\varepsilon)OPT(\tau) - \mathscr{O}((1+\varepsilon)^{-\tau})$ for small $\varepsilon > 0$	bounded support

Table 1: Summary of convergence rate results and necessary assumptions.

We interpret our results as asymptotic properties of the batching and greedy policies with respect to the sojourn period. By instead re-scaling arrival rates and keeping the sojourn period constant, our results also apply to large-market regimes in which the arrival rates increase.

We also extend our model to consider *impatient* nodes, where each unmatched node leaves the system with some type-dependent probability at the end of each period; see Aouad and Saritaç (2022) for a similar model in continuous time, where all agents have the same departure probability. Let $d(\tau)$ denote this departure probability; we show that the results in Table 1 continue to hold for the batching policy if $d(\tau) = \mathscr{O}(\tau^{-\beta})$ for $\beta > 1$, and they continue to hold for the randomized greedy policy if $d(\tau) = o(1)$. In other words, the batching policy is slightly more vulnerable to impatience, while the randomized greedy policy requires less patience to remain asymptotically optimal.

1.2 Related Literature

We briefly review the literature on dynamic matching, starting with the following two papers, which are most closely related to our study. Aouad and Saritaç (2022) study a dynamic stochastic matching problem in continuous time, where nodes arrive according to a Poisson process and their sojourn times are drawn independently from type-specific exponential distributions. Ashlagi et al. (2022) study an online matching problem in discrete time with adversarial arrivals, where vertices have a uniform and fixed sojourn time. Both papers develop heuristic policies with multiplicative worst-case performance guarantees. Our research focuses on an online stochastic matching problem in discrete time with a fixed and uniform sojourn time, and we provide asymptotic performance guarantees for the policies we analyze.

Regarding the batching policy, Aouad and Saritaç (2022) show that it may perform arbitrarily bad in their model by varying certain parameters but keeping the expected sojourn times fixed. Conversely, we show the asymptotic optimality of the batching policy as the sojourn time grows.

Other than these two papers, there have been a few studies related to dynamic non-bipartite matching. Collina et al. (2020) study a similar arrival model to Aouad and Saritaç (2022), and develop a linear programming (LP)-based algorithm, which gives a weaker competitive ratio bound. Ezra et al. (2022) propose prophet inequality algorithms for online weighted matching under vertex and edge arrival settings. Under the edge arrival setting they consider, an edge weight is revealed on arrival and their algorithm decides whether to include the arriving edge in the matching or not. Several research papers focus on the maximum reward matching problem in a fully dynamic graph, in which edges appear and disappear over time (Bhattacharya et al. 2016, 2022, Behnezhad et al. 2020, Neiman and Solomon 2015, Bernstein et al. 2021). Here, the decision maker may change matching decisions as edge insertions and deletions occur. The goal is to design an algorithm with a good approximation ratio and small "update time", the time to process a single edge-change to the graph. For example, Bhattacharya et al. (2022) propose randomized approximation algorithms with poly-logarithmic update time and an approximation ratio strictly better than 2.

Dynamic non-bipartite matching models have also received attention in the queueing literature, where nodes arrive sequentially and wait in queues until being matched; several research papers analyze the performance of greedy policies. Kerimov et al. (2022) proved a greedy longest-queue policy with a minor variation is hindsight-optimal. In finite-horizon multi-way matching, which is motivated by online resource allocation, Gupta (2022) proved the efficacy of the greedy algorithm under certain conditions. Some papers focus on the trade-off between maximizing the total reward and keeping the queues stable. Nazari and Stolyar (2019) proposed an asymptotically optimal matching policy that keeps the queues stable. The models considered in this queueing literature are different from ours in that there is no limit on node waiting time. In our model, we need to decide whether to match a node or let it leave the system when its waiting time reaches a given limit, the sojourn time.

Although we consider a non-bipartite graph, there is extensive literature regarding online bipartite matching, which is also related to our research. The classical setting for online bipartite matching, introduced by Karp et al. (1990), is a system where one side of the bipartition is fixed and known in advance, while nodes from the other side appear sequentially and must immediately be matched or discarded. Karp et al. (1990) studied an adversarial arrival setting, and subsequently Feldman et al. (2009) were the first to consider an i.i.d. stochastic arrival model, in which arriving nodes are drawn repeatedly from a known uniform distribution. As we study stochastic matching in this paper, we restrict our attention to online bipartite stochastic matching. Many papers design a heuristic policy with a performance guarantee relying on an LP relaxation, often with network flow structure (Feldman et al. 2009, Bahmani and Kapralov 2010, Brubach et al. 2016, Manshadi et al. 2012, Jaillet and Lu 2014). Others focus on the derivation of strong relaxations (Torrico et al. 2018, Torrico and Toriello 2022). For additional results, we refer the reader to the survey by Mehta (2013).

In a bipartite matching setting where both sides are dynamic, many studies propose simple policies and analyze their asymptotic properties. When node arrivals follow independent Poisson processes and sojourn times are exponentially distributed, Özkan and Ward (2020) propose a matching policy based on a linear program and Blanchet et al. (2022) propose two types of one-parameter policies, population and utility threshold policies. Aveklouris et al. (2021) consider a weaker assumption that node arrivals follow a type-specific renewal process and sojourn periods follow a type-specific known distribution with bounded supports. They propose *discrete review matching* and *state-independent priority* policies and prove their asymptotic optimality under additional assumptions. Like the approaches used in these papers, we prove the asymptotic optimality of the batching and greedy policies by relying on a fluid relaxation.

2 Problem Description

We consider a discrete-time model with a finite set N representing different *node types* that may appear each period. A node of type $i \in N$ appears in each period following a type-specific independent distribution. Let X_i^t be the random variable for the number of arrivals of type i in period t. For each type i, X_i^t for any $t \in \mathbb{N}$ is independent and identically distributed, and for simplicity we denote by X_i the random variable for the number of arrivals of type i in an arbitrary period; we assume all X_i have finite support unless otherwise noted. Each type remains in the system for a *sojourn period* $\tau \in \mathbb{N}$. That is, a node arriving in period t leaves the system immediately before period $t+\tau$ if it is unmatched. If nodes of type i and j are matched, they yield a reward $w_{ij} \geq 0$ and immediately leave the system; we assume $w_{ij} = 0$ if types i and j are incompatible, and thus we assume the edge set E is complete without loss of generality. In particular, we assume $w_{ii} = 0$ for any type, i.e. two nodes of the same type are incompatible. The objective is to maximize the long-run average reward. Throughout the paper, we use the shorthand $[n] := \{1, \dots, n\}$ for every integer $n \in \mathbb{N}$. For convenience of notation, we use (i, r) to represent a node of type i with r remaining periods, for $i \in N$ and $r \in [\tau]$.

Recall that we evaluate the long-run average reward obtained from batching and greedy policies as a function of the sojourn time τ. We denote the average reward of the batching, greedy and optimal policies respectively as B, G and OPT. Theorem 1 gives our main results.

Theorem 1. *Let $H(\tau)$ be the long-run average reward for policy H, given sojourn time τ. Assuming all X_i^t have bounded support,*

$$
\begin{aligned}
&B(\tau) \geq OPT(\tau) - \mathcal{O}(1/\sqrt{\tau}), \\
&B(\tau) \geq (1-\varepsilon)OPT(\tau) - \mathcal{O}(e^{-\tau\varepsilon}) \text{ for small } \varepsilon > 0, \\
&G(\tau) \geq OPT(\tau) - \mathcal{O}(1/\tau), \\
&G(\tau) \geq (1-\varepsilon)OPT(\tau) - \mathcal{O}((1+\varepsilon)^{-\tau}) \text{ for small } \varepsilon > 0.
\end{aligned}
$$

In Sections 3 and 4, we provide the proofs for Theorem 1. They rely on the analysis for a single-pair case and the conversion from the single-pair case to the general counterpart using an LP relaxation and randomization. Before the proofs, we formally describe our problem in what follows.

2.1 Markov Decision Process Formulation

We present an infinite-horizon average reward Markov decision process (MDP) formulation for the dynamic matching problem. Let S_i^r be the number of nodes of type i with r remaining periods in the system. We define state S by a sequence of S_i^r for $i \in N$ and $r \in [\tau]$. To ease notation, we use $S = (S^\tau, \ldots, S^1)$, where S^r is a subsequence $\{S_i^r\}_{i\in N}$. We denote the state space by $\mathscr{S}$, which is finite when all X_i^t have bounded support. A matching action x is an integer vector satisfying matching constraints, namely that an edge incident to a node of type i with r remaining periods can be used up to S_i^r times. The action space at state S is $\mathscr{X}(S) = \{x \in \mathbb{Z}_+^E : \sum_{e\in\delta(i,r)} x_e \le S_i^r,\ i \in N, r \in [\tau]\}$. We denote by $S^r(x)$ the post-decision vector of nodes that are not matched by x, i.e. $S_i^r(x) = S_i^r - \sum_{e\in\delta(i,r)} x_e$. At the end of each period, the number of remaining periods of each unmatched node decreases by 1; if this becomes 0, the node immediately leaves the system. Subsequently, nodes with τ remaining periods appear following the corresponding arrival distributions. That is, for each $S, U \in \mathscr{S}$ and $x \in \mathscr{X}(S)$, the transition probability $\mathbb{P}(U|S,x)$ is $\phi(U^\tau) = \prod_{i\in N} \mathbb{P}(X_i = U_i^\tau)$ if $U^r = S^{r+1}(x)$, $r \in [\tau-1]$, and equals zero otherwise. The reward for action x at state S is the sum of its matching rewards, $\sum_{e\in E} w_e x_e$.

Since this MDP is communicating and has a finite state space, the optimal expected average reward is constant (Puterman 2014). We denote the expected average reward and bias respectively as η and $v(S)$; the LP formulation for the model is

$$\left\{\min\ \eta : \eta + v(S) - \sum \phi(R) v(R, S^\tau(x), \ldots, S^2(x)) \ge \sum_{e\in E} w_e x_e, \quad S \in \mathscr{S}, x \in \mathscr{X}(S)\right\}, \tag{1}$$

where R represents the vector of new arrivals. The dual of (1) is

$$\max_{z\ge 0} \sum_{S\in\mathscr{S}, x\in\mathscr{X}(S)} z(S,x) \sum_{e\in E} w_e x_e$$

$$\text{s.t.} \sum_{y\in\mathscr{X}(U)} z(U,y) - \sum_{S\in\mathscr{S}, x\in\mathscr{X}(S):\ U^r = S^{r+1}(x)\ \forall r\in[\tau-1]} \phi(U^\tau) z(S,x) = 0, \quad U \in \mathscr{S} \tag{2a}$$

$$\sum_{S\in\mathscr{S}, x\in\mathscr{X}(S)} z(S,x) = 1, \tag{2b}$$

where $z(S,x)$ encodes the probability of arriving at state S and choosing action x in any given period.

2.2 Linear Programming Relaxation

Next, we construct a fluid relaxation of the MDP, which we use as a benchmark. We begin by presenting a property of this problem.

Proposition 2. *There exists an optimal deterministic policy that matches nodes only if at least one of them is about to leave the system, i.e. it only has one remaining period.*

Proof. By standard results in Markov decision processes, the problem has an optimal deterministic policy (Puterman 2014). Let π be an arbitrary deterministic optimal policy. Suppose there exists a state S such that the optimal matching decision at S following π contains pairs with no expiring node. Denote such pairs by M. For any sample path, whenever S appears, it is possible to modify the path by postponing each pair in M until one of its nodes is about to leave. Since this does not change the average reward, the policy following the modified sample path is optimal. This modification can be repeated as necessary until π satisfies the claimed property. □

By Proposition 2, we can assume that nodes are matched only if at least one of them is about to leave the system. Based on this additional assumption, we can assume that E consists only of edges incident to expiring nodes. By introducing aggregating variables and rewriting the constraints in (2), we construct an LP formulation, which we utilize as a benchmark throughout this paper:

$$\max_{z\geq 0}\left\{\sum_{i,j\in N} w_{ij}z_{ij} : \sum_{j\in N} z_{ij} \leq \mathbb{E}[X_i], \quad i\in N\right\}. \tag{3}$$

Proposition 3. *The LP (3) is a relaxation of (2).*

We defer the proof of the proposition to Appendix A.1.

3 Convergence Analysis for a Single Pair

Next, we show convergence results for a single node pair $\{i,j\}$.

3.1 Batching Policy

For a single node pair $\{i,j\}$, the batching policy has a long-term matching frequency of $\mathbb{E}[\min\{A_i,A_j\}]/\tau$, where A_i is the number of type i arrivals within the batching interval τ, $A_i = \sum_{t=1}^{\tau} X_i^t$.

Proposition 4. *Let A_i and A_j respectively be the sum of τ independent and identically distributed random variables with finite variance, $A_i = \sum_{t=1}^{\tau} X_i^t, A_j = \sum_{t=1}^{\tau} X_j^t$. Then,*

$$\lim_{\tau\to\infty} \frac{\mathbb{E}[\min\{A_i,A_j\}]}{\tau} = \min\{\mathbb{E}[X_i],\mathbb{E}[X_j]\}.$$

The gap between the left and right-hand side is $\mathscr{O}(1/\sqrt{\tau})$, and may be $\Theta(1/\sqrt{\tau})$, for example when X_i and X_j are both Bernoulli random variables with probability $1/2$. When $\mathbb{E}[X_i] < \mathbb{E}[X_j]$ and both random variables have finite support, the gap is $\mathscr{O}(e^{-\tau(\mathbb{E}[X_j]-\mathbb{E}[X_i])})$.

Proof. Without loss of generality, assume $\mathbb{E}[X_i] \le \mathbb{E}[X_j]$; then,

$$\begin{aligned}
\mathbb{E}[|A_i - A_j|] &\le \mathbb{E}[|A_i - A_j - \mathbb{E}[A_i - A_j]|] + \mathbb{E}[|\mathbb{E}[A_i - A_j]|] \\
&\le \sqrt{\mathbb{E}[|A_i - A_j - \mathbb{E}[A_i - A_j]|^2]} + (\mathbb{E}[A_j] - \mathbb{E}[A_i]) \\
&= \sqrt{\mathrm{Var}[A_i - A_j]} + (\mathbb{E}[A_j] - \mathbb{E}[A_i]).
\end{aligned}$$

The first inequality follows from the triangle inequality, and the second from Jensen's inequality. We then get

$$\begin{aligned}
\mathbb{E}[\min\{A_i,A_j\}] &= \frac{1}{2}(\mathbb{E}[A_i + A_j] - \mathbb{E}[|A_i - A_j|]) \\
&\ge \mathbb{E}[A_i] - \frac{1}{2}\sqrt{(\mathrm{Var}[A_i - A_j])} = \tau\mathbb{E}[X_i] - \frac{1}{2}\sqrt{\tau(\mathrm{Var}[X_i] + \mathrm{Var}[X_j])}.
\end{aligned}$$

We defer the proof for the existence of an instance where the convergence rate is tight to Appendix A.3.

Finally, we prove that the convergence rate is $\mathscr{O}(e^{-\tau(\mathbb{E}[X_j]-\mathbb{E}[X_i])})$ when $\mathbb{E}[X_i] < \mathbb{E}[X_j]$. Since X_i and X_j have bounded supports, there exists a finite $m \in \mathbb{N}$ such that $\mathbb{P}(X_j - X_i \le m) = 1$. We first derive an upper bound on the expectation of the positive components of $A_i - A_j$,

$$\begin{aligned}
\mathbb{E}[(A_i - A_j)1_{\{A_j - A_i < 0\}}] &\le \mathbb{E}[\tau m 1_{\{A_j - A_i < 0\}}] = \tau m \mathbb{P}(A_j - A_i < 0) \\
&= \tau m \mathbb{P}\Big(A_j - A_i - \tau(\mathbb{E}[X_j] - \mathbb{E}[X_i]) < -\tau(\mathbb{E}[X_j] - \mathbb{E}[X_i])\Big) \\
&\le 2\tau m \exp\Big(-\frac{2\tau}{m^2}(\mathbb{E}[X_j] - \mathbb{E}[X_i])^2\Big),
\end{aligned}$$

where the first inequality is by the definition of m and the second follows from Hoeffding's inequality. Using this, we get

$$\mathbb{E}[|A_j - A_i|] = \mathbb{E}[(A_j - A_i) - 2(A_j - A_i)1_{\{A_j - A_i < 0\}}] = \mathbb{E}[(A_j - A_i)] + 2\mathbb{E}[(A_i - A_j)1_{\{A_j - A_i < 0\}}]$$
$$\leq \mathbb{E}[(A_j - A_i)] + 4\tau m \exp\Big(-\frac{2\tau}{m^2}(\mathbb{E}[X_j] - \mathbb{E}[X_i])^2\Big).$$

Finally, it follows that

$$\mathbb{E}[\min\{A_i, A_j\}] = \frac{1}{2}\Big(\mathbb{E}[A_i + A_j] - \mathbb{E}[|A_j - A_i|]\Big) \geq \mathbb{E}[A_i] - 2\tau m \exp\Big(-\frac{2\tau}{m^2}(\mathbb{E}[X_j] - \mathbb{E}[X_i])^2\Big). \qquad \square$$

By Proposition 4, the long-run average matching frequency of a pair $\{i, j\}$ following the batching policy converges to the optimal solution of the LP (3) with an optimality gap of $\mathscr{O}(1/\sqrt{\tau})$. The convergence is exponentially fast when $\mathbb{E}[X_i] < \mathbb{E}[X_j]$, which allows us to establish exponential convergence to $(1-\varepsilon)$-optimality: if $\mathbb{E}[X_i] \leq (1-\varepsilon)\mathbb{E}[X_j]$, then batching converges to optimality with $\mathscr{O}(e^{-\tau\varepsilon})$ gap; otherwise, we randomly discard every arrival of type i independently with probability ε before applying the batching policy. This random discarding reduces the expected number of type i arrivals to no more than $(1-\varepsilon)\mathbb{E}[X_j]$, which ensures convergence with gap $\mathscr{O}(e^{-\tau\varepsilon})$ to $(1-\varepsilon)\mathbb{E}[X_i] = (1-\varepsilon)\min\{\mathbb{E}[X_i], \mathbb{E}[X_j]\}$.

In addition, there exist instances where this gap is tight. In the next section, we show that the long-run average matching frequency of a pair $\{i, j\}$ following the greedy policy converges to the optimum with an optimality gap of $\mathscr{O}(1/\tau)$. We conclude that there are instances where the batching policy converges to the optimum with an optimality gap of $\Theta(1/\sqrt{\tau})$.

3.2 Greedy Policy

We will find a closed-form equation for the long-run average matching frequency of a pair $\{i, j\}$ following the greedy policy. In the case of a single pair, the greedy policy is simple: execute a match whenever possible; if multiple nodes of one type are available to match, choose the node that has waited the longest.

We analyze a discrete-time Markov chain with state defined to be the sequence of the number of (node type, remaining period) combinations in the system, a Markov reward process where we apply the greedy policy. If this chain is ergodic and has a stationary distribution π, we can compute the long-run average matching frequency of $\{i, j\}$ as $\sum_{S\in\mathscr{S}} \pi(S)\alpha(S)$, where $\alpha(S)$ is the number of matches made in state S following the greedy policy. We first present some properties of the chain, show its ergodicity and the existence of the stationary distribution. Next, we establish a transition diagram; based on the transition diagram, we propose a weight function satisfying the detailed balance equations. By normalizing the weight function, we obtain a stationary distribution, and compute the long-run average matching frequency for $\{i, j\}$.

For this analysis, we first assume node type arrivals follow Bernoulli distributions, $X_i \sim \text{Bernoulli}(p_i)$, $X_j \sim \text{Bernoulli}(p_j)$ for $p_i, p_j \in (0,1]$. Under this assumption, we use a simplified state notation, $S = (S_i^\tau, \ldots, S_i^1, S_j^\tau, \ldots, S_j^1)$, where $S_i^r, S_j^r \in \{0,1\}$ for $r \in [\tau]$. We begin by presenting properties of the Markov chain resulting from a Markov reward process with the greedy policy.

Lemma 5. *For $S \in \mathscr{S}$, if there exists $r \in [\tau - 1]$ with $S_i^r = 1$, then $S_j^t = 0\ \forall t \in [\tau - 1]$, and vice versa.*

Proof. Suppose not; then, there exist $r, t \in [\tau - 1]$ such that $S_i^r = 1, S_j^t = 1$, implying the previous state had an available pair $S_i^{r+1} = 1$ and $S_j^{t+1} = 1$ that was not matched, a contradiction. □

We note that Lemma 5 does not say anything about period τ. That is, even if there exists $r \in [\tau - 1]$ such that $S_i^r = 1$, S_j^τ can be 1 when there is a new node arrival of type j. This newly arrived node of type j is matched with any existing node of type i (if one exists) and both leave the system at the end of the period. Afterwards, the remaining period of each remaining node is decreased by 1. Lastly, new node arrivals possibly occur. Figure 1 depicts an example of this process. Let $f(S)$ be the post-state of S, the state immediately before new arrivals occur. That is, $f(S) = (f_i(S), f_j(S))$, where $f_i(S)$ is a sequence of length $(\tau - 1)$ for type i, and similarly for type j. At the beginning of a period, the policy matches available pairs, colored in gray in the figure. Matched nodes leave the system, and the remaining period for each node still in the system decreases by 1. The post-state is defined with these nodes, colored blue and green in the figure. The next state is then defined by adding any newly arrived nodes.

Notice that $f(S)$ is deterministic for a given state S, and the randomness of the next state arises from the newly arrived node types. We formally describe this state transition in the following lemma.

Lemma 6. *For each state $S \in \mathscr{S}$, the next state is $(a, f_i(S), b, f_j(S))$ with probability $\mathbb{P}(X_i = a)\mathbb{P}(X_j = b)$ for $a, b \in \{0,1\}$.*

Proof. Since $f(S)$ is deterministic, the state transition probability only depends on the probability of new arrivals. □

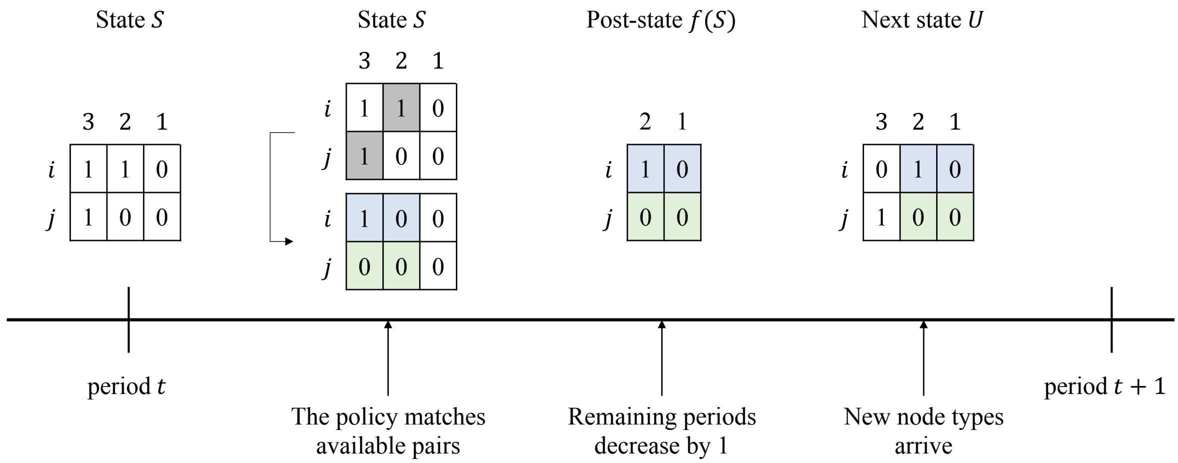

Figure 1: An instance of a transition diagram when $\tau = 3$.

Lemma 7. *The Markov chain resulting from a Markov reward process following the greedy policy is irreducible and aperiodic.*

Proof. We show that any state in $\mathscr{S}$ is communicating with the empty state, which is a zero sequence of length 2τ. Let $S \in \mathscr{S}$ be an arbitrary state. The transition probability from S to the empty state is greater than or equal to $\left(\mathbb{P}(X_i = 0)\mathbb{P}(X_j = 0)\right)^{\tau}$, the probability that neither type i nor j arrive for τ periods. Since this value is positive, the empty state is reachable from S.

The transition probability from the empty state to S is greater than or equal to $\prod_{r\in[\tau]} \mathbb{P}(X_i = S_i^r)\mathbb{P}(X_j = S_j^r)$, the probability that nodes corresponding to S appear. This value is also positive, which implies S is reachable from the empty state.

The period of the empty state is 1, because the probability of its self loop is greater than or equal to $\mathbb{P}(X_i = 0)\mathbb{P}(X_j = 0)$, which is positive. Thus, this chain is aperiodic. □

Since the state space $\mathscr{S}$ is finite and the chain is irreducible, there exists a unique stationary distribution. In addition, by the aperodicity of the chain, the limiting distribution exists and converges to the stationary distribution. Therefore, by solving detailed balance equations, we can find the stationary distribution. To do so, we need to clearly identify the transitions among states. We draw a sample partial transition diagram in Figure 2.

Recall that given state S, its next state depends on post-state $f(S)$ and new arrivals. This implies that a set of states with the same post-state follow the same transition. For example, in Figure 2 the transition probabilities from the states drawn on the leftmost side to state $(1,1,0,1,0,0)$ all equal $p_i p_j$. The states on

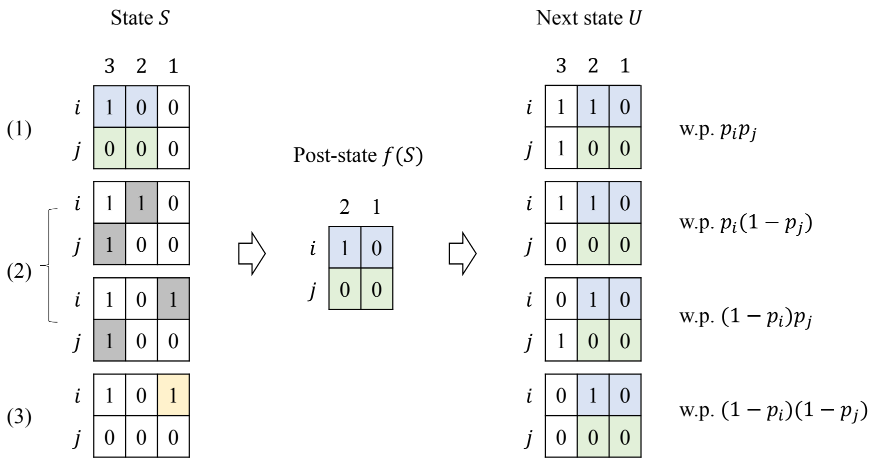

Figure 2: Sample transition diagram when $\tau = 3$.

the leftmost side consist of three groups: Group (1) is a singleton set with the state constructed by increasing the remaining period of the nodes in the post-state by 1 and assuming no nodes with remaining period 1, i.e. $(f_i(S), 0, f_j(S), 0)$. Group (2) is a set of states in which a match with a newly arrived node of type j and a previously existing node of type i occurs; the nodes colored in gray represent matches. Group (3) is a singleton set with the state constructed by increasing the remaining period of nodes in the post-state by 1 and adding a node of type i with remaining period 1 which leaves the system unmatched, i.e. $(f_i(S), 1, f_j(S), 0)$; the node colored in yellow leaves the system unmatched. We formally describe the transitions and propose a function satisfying detailed balance equations in Proposition 8.

Proposition 8. *Define a function $h : \mathscr{S} \to \mathbb{R}$ with*

$$h(S) = \mathbb{P}(X_i^\tau = S_i^\tau)\mathbb{P}(X_j^\tau = S_j^\tau) \times \begin{cases} (1-p_j)^{\tau - r(S_i)}(\frac{p_i}{1-p_i})^{|T(S_i)|} & \text{if } T(S_i) \neq \emptyset, T(S_j) = \emptyset \\ (1-p_i)^{\tau - r(S_j)}(\frac{p_j}{1-p_j})^{|T(S_j)|} & \text{if } T(S_i) = \emptyset, T(S_j) \neq \emptyset \\ 1 & \text{if } T(S_i) = \emptyset, T(S_j) = \emptyset, \end{cases}$$

where $r(S_i) = \min\{t \in [\tau - 1] : S_i^t = 1\}$ is the smallest remaining period of a node of type i in S, $T(S_i) = \{t \in [\tau - 1] : S_i^t = 1\}$ is the set of remaining periods of nodes of type i in S, and $r(S_j)$ and $T(S_j)$ are defined analogously. This h satisfies the Markov chain's balance equations.

Note that there is no state with $T(S_i) \neq \emptyset$ and $T(S_j) \neq \emptyset$, by Lemma 5.

Proof. Let S be an arbitrary state in $\mathscr{S}$. Let $g(S)$ be the subsequence of S consisting of node types with remaining period less than τ. That is, $g(S) = (g_i(S), g_j(S))$ where $g_i(S) = (S_i^{\tau-1}, \ldots, S_i^1)$. Following from Lemma 6, the transition probability from U to S is $\mathbb{P}(X_i^\tau = S_i^\tau)\mathbb{P}(X_j^\tau = S_j^\tau)$ if $f(U) = g(S)$, and zero otherwise. In addition, by Lemma 5, $g_i(S)$ or $g_j(S)$ is a zero sequence.

We first consider the case when both $g_i(S)$ and $g_j(S)$ are zero sequences. The list of states U such that $f(U) = g(S)$ is

1. $U_i = (0,0,\ldots,0)$, $U_j = (0,0,\ldots,0)$
2. $U_i = (0,0,\ldots,0) + e_t$, $U_j = (1,0,\ldots,0)$ for $t \in [\tau]$
3. $U_i = (1,0,\ldots,0)$, $U_j = (0,0,\ldots,0) + e_t$ for $t \in [\tau-1]$
4. $U_i = (0,0,\ldots,1)$, $U_j = (0,0,\ldots,0)$
5. $U_i = (0,0,\ldots,0)$, $U_j = (0,0,\ldots,1)$,

where e_t is the t-th standard vector, e.g. $e_1 = (0,0,\ldots,1)$. In case (2), a node of type i with remaining period t is matched with a newly arrived node of type j. Case (3) is the reverse; we exclude $t = \tau$ in case (3) because it is already included in case (2). In case (4), a node of type i with remaining period 1 leaves the system unmatched, and case (5) is the reverse. Plugging in function h, the in-flow amount to S is

$$\mathbb{P}(X_i^\tau = S_i^\tau)\mathbb{P}(X_j^\tau = S_j^\tau)\Big[(1-p_i)(1-p_j) + (1-p_i)p_j \sum_{t\in[\tau]} (1-p_j)^{\tau-t}\Big(\frac{p_i}{1-p_i}\Big) + p_i(1-p_j) \sum_{t\in[\tau-1]} (1-p_i)^{\tau-t}\Big(\frac{p_j}{1-p_j}\Big) + (1-p_i)(1-p_j)(1-p_j)^{\tau-1}\Big(\frac{p_i}{1-p_i}\Big) + (1-p_i)(1-p_j)(1-p_i)^{\tau-1}\Big(\frac{p_j}{1-p_j}\Big)\Big].$$

We show that the summation in the square brackets is equal to 1 in Appendix A.5. Then, it follows that the in-flow amount to S is $\mathbb{P}(X_i^\tau = S_i^\tau)\mathbb{P}(X_j^\tau = S_j^\tau)$. Since we are considering the case when $g_i(S)$ and $g_j(S)$ are zero sequences, $h(S) = \mathbb{P}(X_i^\tau = S_i^\tau)\mathbb{P}(X_j^\tau = S_j^\tau)$. Thus, we conclude that the in-flow amount to S is $h(S)$.

We next assume that $g_i(S)$ is not a zero sequence. The list of states U such that $f(U) = g(S)$ is

1. $U_i = (g_i(S), 0)$, $U_j = (0,0,\ldots,0)$
2. $U_i = (g_i(S), 0) + e_t$, $U_j = (1,0,\ldots,0)$ for $t \in [r(S_i)]$
3. $U_i = (g_i(S), 1)$, $U_j = (0,0,\ldots,0)$.

In case (2), a node of type i with remaining period t is matched with a newly arriving node of type j. In case (3), a node of type i leaves the system unmatched. Plugging in function h, we can verify that the in-flow amount to S is equal to $h(S)$ following steps similar to the previous argument. The same reasoning applies when $g_j(S)$ is not a zero sequence. □

We obtain the stationary distribution π by normalizing h, since $\mathscr{S}$ is finite. Using π, we can compute the long-run average matching frequency of pair $\{i,j\}$.

Proposition 9. *The long-run average matching frequency of pair $\{i,j\}$ under the greedy policy is*

$$\begin{cases} p - \frac{p(1-p)}{1+2(\tau-1)p} & \text{if } p_i = p_j = p \\ p_i - \frac{(p_j-p_i)p_i(1-p_i)}{p_j(1-p_j)q^{-2\tau}-p_i(1-p_i)} & \text{if } p_i < p_j, q = \frac{1-p_j}{1-p_i} < 1. \end{cases}$$

Proof. Let $S=(S_i,S_j)$ be an arbitrary state in $\mathscr{S}$. By Lemma 5, there are three possible cases:

1. $\mathscr{S}_i : T(S_i) \neq \emptyset, T(S_j) = \emptyset$
2. $\mathscr{S}_j : T(S_i) = \emptyset, T(S_j) \neq \emptyset$
3. $\mathscr{S}_0 : T(S_i) = \emptyset, T(S_j) = \emptyset$.

In case (1) (case (2)), the decision maker executes one match if $S_j^\tau = 1$ ($S_i^\tau = 1$). In case (3), the decision maker executes one match if $S_j^\tau = 1$ and $S_i^\tau = 1$. Thus, the long-run average matching frequency is

$$\mathbb{P}(S_j^\tau = 1)\sum_{S\in\mathscr{S}_i}\pi(S) + \mathbb{P}(S_i^\tau = 1)\sum_{S\in\mathscr{S}_j}\pi(S) + \mathbb{P}(S_i^\tau = 1)\mathbb{P}(S_j^\tau = 1)\sum_{S\in\mathscr{S}_0}\pi(S),$$

which is equivalent to

$$\frac{1}{\sum_{S\in\mathscr{S}} h(S)}\left[p_j \sum_{S\in\mathscr{S}_i} h(S) + p_i \sum_{S\in\mathscr{S}_j} h(S) + p_i p_j \sum_{S\in\mathscr{S}_0} h(S)\right].$$

By plugging in h and rearranging the terms, we obtain the equations in the proposition. We present the details in Appendix A.6. □

As a result, the long-run average matching frequency is $p_i - \mathscr{O}(1/\tau)$ when $p_i = p_j$, and $p_i - \mathscr{O}((\frac{1-p_j}{1-p_i})^\tau)$ when $p_i < p_j$. The latter result allows us to establish the exponential convergence to $(1-\varepsilon)$-optimality, in a similar fashion to the batching policy.

We generalize the convergence result for Bernoulli distributions to general discrete distributions with bounded support by using randomization of Markov chains. Let m be the maximum number of type arrivals per period, i.e. $\mathbb{P}(X_i \le m) = \mathbb{P}(X_j \le m) = 1$. Roughly speaking, we generate m copies of a Bernoulli Markov "sub-chain" and randomly distribute a type's arrivals among the m chains. Let S_k be the state of chain k for $k \in [m]$. By defining state S as $\{S_k\}_{k\in[m]}$, which is the sequence of states of the m chains, we can convert the original Markov chain into a new chain with larger state space; we call this chain the *converted* chain. Once new types arrive, we assume each sub-chain operates independently. That is, we do not match types in different sub-chains.

Let $X_{i,k}$ be the random variable for the number of new arrivals of type i to chain k for any $k \in [m]$. By construction, $\mathbb{P}(X_{i,1} = n_1, X_{i,2} = n_2, \ldots, X_{i,m} = n_m | X_i = n) = \binom{m}{n}^{-1}$ if $\sum_{k=1}^m n_k = n$ and $n_k \in \{0,1\}$ for $k \in [m]$; the probability is zero otherwise. This implies $\mathbb{P}(X_{i,k} = 1 | X_i = n) = \binom{m-1}{n-1} / \binom{m}{n} = n/m$, and the marginal probability is then $\mathbb{P}(X_{i,k} = 1) = \sum_{n=1}^m \mathbb{P}(X_i = n) n/m = \mathbb{E}[X_i]/m \le 1$. This yields a Markov chain with Bernoulli random arrivals, where type i and j arrive with probability $\mathbb{E}[X_i]/m$ and $\mathbb{E}[X_j]/m$, respectively, and we denote it the *Bernoulli* chain. In the next proposition, we prove that the long-run average matching frequency of the converted chain equals m times the long-run average matching frequency of the Bernoulli chain, and provides a lower bound for the original long-run average matching frequency.

Proposition 10. *Let z^*, z^c, and z^B be the long-run average matching frequency of the original, converted, and Bernoulli chains, respectively. Then, $z^* \ge z^c = m z^B$.*

Proof. The inequality is trivial because our assumption that types in different sub-chains cannot be matched means the converted chain only misses potential matches. To show the equality, we first define the stationary distributions of the given chains. Let π^c and π^B be the stationary distributions of the converted and Bernoulli chains, respectively. We claim that it suffices to show that $\sum_{k\in[m]\setminus\{\ell\}} \sum_{S_k} \pi^c(S_1, \ldots, S_m) = \pi^B(S_\ell)$ for any $\ell \in [m]$. Suppose so; let $\alpha^B(S)$ and $\alpha^c(S)$ be the number of matches that the decision maker executes in state S of the Bernoulli and converted chains, respectively, where S is defined corresponding to each chain. Similarly, we define $\alpha_k^c(S)$ to be the number of matches in state S of sub-chain k for $k \in [m]$. Since sub-chain k and the Bernoulli chain have the same probability distribution of new type arrivals and the same transition mechanism, $\alpha^B(S) = \alpha_k^c(S)$ for any state S and $k \in [m]$. Therefore,

$$\begin{aligned}
z^c &= \sum_{k\in[m]} \sum_{S_k} \pi^c(S_1,\dots,S_m)\alpha^c(S_1,\dots,S_m) \\
&= \sum_{k\in[m]} \sum_{S_k} \pi^c(S_1,\dots,S_m)\Big(\alpha_1^c(S_1)+\cdots+\alpha_m^c(S_m)\Big) \\
&= \sum_{\ell\in[m]} \alpha_\ell^c(S_\ell) \sum_{k\in[m]\setminus\{\ell\}} \pi^c(S_1,\dots,S_m) \\
&= \sum_{\ell\in[m]} \alpha_\ell^c(S_\ell)\pi^B(S_\ell) = \sum_{\ell\in[m]} z^B = mz^B,
\end{aligned}$$

where the second equality follows from the assumption that we do not match types in different sub-chains. We defer the proof of the claim to Appendix A.7. □

4 Convergence Analysis for the General Case

For the general case, we convert the original problem into separate single-pair instances by introducing the concept of a *sub-type*. We first present the conversion logic based on randomization and verify that randomized policies are asymptotically optimal. We then show that the batching policy dominates its randomized counterpart. In addition, we describe a modified greedy policy that is more practical than the randomized greedy policy while inheriting its performance guarantee.

4.1 Randomization

Recall the fluid relaxation (3), where z_{ij} is the long-run matching frequency for $\{i,j\}$. We denote the optimal solution by z_{ij}^*, and for any $z_{ij}^* > 0$ we define $\bar{z}_{ij}^* := z_{ij}^* / \sum_{k\in N\setminus i} z_{ik}^*$ as the normalized value, which is asymmetric. We randomly assign arriving nodes of type i to sub-type (i,j) with probability $\bar{z}_{ij}^*$. Since the sub-type assignment and type arrivals are independent, the random variable for the number of arrivals of type i with sub-type (i,j), denoted by X_i^j, is a random variable with $\mathbb{E}[X_i^j] = \bar{z}_{ij}^*\mathbb{E}[X_i]$. From the single-pair results, the long-run matching frequency for $\{i,j\}$ under either policy converges to $\min\{\mathbb{E}[X_i]\bar{z}_{ij}^*, \mathbb{E}[X_j]\bar{z}_{ji}^*\} \geq z_{ij}^*$. Using the optimality of z^* for the relaxation, it follows that the randomized policies are asymptotically optimal.

4.2 Policy Comparison

The batching policy dominates its randomized counterpart as defined above; in any batch, the randomized policy's matching is a feasible solution for that batch's max-reward matching problem.

In contrast, it is not straightforward to show that a greedy policy without randomization dominates its randomized counterpart. Consider the following example with three node types i, j, k. Let $w_{ij} = w_{ik} = \delta$ for small $\delta > 0$, $w_{jk} = 1$, $p_i = 1$, $p_j = p_k = p < 1$. By Proposition 9, the policy that matches types j and k whenever possible and ignores type i has a long-run average reward that converges to p as τ grows. On the other hand, a straightforward greedy policy has a matching frequency for this pair of p^2, because if a node of type j or k appears without the other, the policy will match it to a node of type i.

With this motivation, we propose a *modified delayed greedy* policy that dominates the *randomized greedy* policy and thus inherits its performance guarantees. In each period, we begin with a matching inherited from the end of the previous period; recall that we can delay our decisions as late as possible by Proposition 2, meaning we only execute matches when one of the nodes is about to leave the system, and thus any other potential matches are not executed and stay in the system. If any newly arriving nodes can form new matches according to their sub-types, we add these to the matching. For each matched pair $\{(i,r),(j,t)\}$, if there exists an unmatched node of type i or j with a smaller remaining period in the system, we swap that node with the current node. When we perform such a swap, we swap not only the match but also the sub-type. That is, if there is a node of type i with remaining period $r' < r$, we now match $\{(i,r'),(j,t)\}$ instead of $\{(i,r),(j,t)\}$, and set the sub-type of (i,r) as the original sub-type of (i,r'). We repeat this procedure until there are no possible swaps. Let M be the matching after this procedure. Considering only nodes covered by M and nodes that are unmatched and about to leave the system, we solve a max-reward matching problem. We execute matches where the matched pair contains a node that is about to leave the system; the remaining matching edges remain in the system for the next period, and the process repeats. This modified delayed greedy policy dominates the randomized greedy policy because swapping nodes does not change the total reward and reoptimizing only increases the total reward; thus the modified policy inherits the performance guarantees of its randomized counterpart.

5 Impatience

In this section, we extend our results to the setting in which unmatched nodes may leave the system before the sojourn time elapses. Each unmatched node of type $i \in N$ abandons the system with probability $d_i(\tau)$ in each period; we explicitly consider the departure probability as a function of τ. Intuitively, we can interpret this condition as indicating that the system manager guarantees arriving agents a waiting time no longer than τ, but agents are impatient and may leave the system before being matched; however, the knowledge of τ influences the agents' patience. The departure probabilities $d_i(\tau)$ may differ by type, but we assume they change with the same rate with respect to τ. In Theorems 11 and 12, we characterize how much impatience the batching and greedy policies can accommodate while remaining asymptotically optimal with respect to τ. Specifically, under these conditions the policies' long-run average rewards converge to the optimal objective value of the relaxation (3) with the same rate as when there is no impatience.

Theorem 11. *Assume all X_i^t have bounded support. If $d_i(\tau) = \mathscr{O}(\tau^{-\beta})$ for all $i \in N$ and some $\beta > 1$, the batching policy achieves the same convergence results as in Theorem 1. If $\beta \leq 1$, there exist instances where the batching policy is not asymptotically optimal.*

In particular, this theorem extends a negative result in Aouad and Saritaç (2022) for the continuous time case, which shows that the batching policy is sub-optimal when the departure probabilities are constant.

Proof. The proof follows the same arguments from Theorem 1. We provide a sketch here and present the detailed proof in Appendix A.8. Without loss of generality, we assume a single type pair $\{i, j\}$ with $\mathbb{E}[X_i] \leq \mathbb{E}[X_j]$. For the batching policy, we define Y_i^t as the number of type i nodes that remain in the system for t periods. The number of remaining type i nodes within the batching interval τ, say A_i, is then $\sum_{t=1}^{\tau-1} Y_i^t$. The long-run average matching frequency following the batching policy is $\mathbb{E}[\min\{A_i, A_j\}]/\tau$. By using the proof of Proposition 4 and the definition of Y_i^t, the lower and upper bounds of the long-run average matching frequency following the batching policy are

$$\mathbb{E}[X_i]\frac{1-(1-d_i(\tau))^\tau}{\tau d_i(\tau)} - \mathscr{O}(1/\sqrt{\tau}) \quad \text{and} \quad \mathbb{E}[X_i]\frac{1-(1-d_i(\tau))^\tau}{\tau d_i(\tau)},$$

respectively. If $d_i(\tau) = \mathscr{O}(\tau^{-\beta})$ for $\beta > 1$, both bounds converge to $\mathbb{E}[X_i]$, which implies the long-run average matching frequency following the batching policy converges to $\mathbb{E}[X_i]$.

We next show that if $d_i(\tau) = \mathscr{O}(\tau^{-\beta})$ for $\beta \leq 1$, there exists an instance where the batching policy is not optimal. Suppose $d_i(\tau) = \Theta(\tau^{-1})$ and node type arrivals follow Bernoulli distributions. Recall the lower and upper bounds of the long-run average matching frequency of the batching policy: these converge to a value smaller than p_i, e.g. $(1-1/e)p_i$ if $d_i(\tau) = \tau^{-1}$. On the other hand, under the same settings we can show that the greedy policy converges to p_i. $\square$

Theorem 12. *Assume all X_i^t have bounded support. If $d_i(\tau) = o(1)$ for all $i \in N$, the randomized greedy policy achieves the same convergence results as in Theorem 1. If $d_i(\tau) = \Omega(1)$ for some $i \in N$, there exist instances where the randomized greedy policy's average reward does not converge to the optimal objective value of* (3).

While the randomized greedy policy may not achieve the relaxation's average reward when departure probabilities do not decay, it could still be asymptotically optimal. Showing this would presumably require a stronger relaxation, perhaps using techniques similar to Aouad and Saritaç (2022).

Proof. Following Proposition 10, it suffices to show that the result is true for Bernoulli random arrivals. We assume node type arrivals follow Bernoulli distributions, $X_i \sim \text{Bernoulli}(p_i)$, $X_j \sim \text{Bernoulli}(p_j)$ for $p_i, p_j \in (0,1]$. We verify the following function $h : \mathscr{S} \to \mathbb{R}$ satisfies the Markov chain's balance equations in the presence of impatience:

$$h(S) = \mathbb{P}(X_i^\tau = S_i^\tau)\mathbb{P}(X_j^\tau = S_j^\tau) \times \begin{cases} (1-p_j)^{\tau - r(S_i)} \prod_{t \in T(S_i)} \frac{(1-d_i(\tau))^{\tau-t} p_i}{1-(1-d_i(\tau))^{\tau-t} p_i} & \text{if } T(S_i) \neq \emptyset, T(S_j) = \emptyset \\ (1-p_i)^{\tau - r(S_j)} \prod_{t \in T(S_j)} \frac{(1-d_j(\tau))^{\tau-t} p_j}{1-(1-d_j(\tau))^{\tau-t} p_j} & \text{if } T(S_i) = \emptyset, T(S_j) \neq \emptyset \\ 1 & \text{if } T(S_i) = \emptyset, T(S_j) = \emptyset. \end{cases}$$

By normalizing h, we can compute the stationary distribution and then the long-run average matching frequency following the greedy policy. By using Fatou's lemma, we show that the long-run average matching frequency following the greedy policy converges to $\min\{p_i, p_j\}$ if $d_i(\tau) = d_j(\tau) = o(1)$. We provide the detailed proof in Appendix A.9.

Now suppose $p_i = p_j = p \in (0,1)$, and $d_i(\tau) = d_j(\tau) = 1 - \varepsilon$ for a small $\varepsilon > 0$. Then the average reward converges to p^2 as $\varepsilon \to 0$, because the greedy policy can only execute matches when nodes from both types appear simultaneously. We present a rigorous proof of this argument in Appendix A.9. $\square$

6 Experimental Study

Our theoretical results show that batching and greedy policies converge to optimality as the sojourn time increases. In this section, we use empirical simulations to show that these policies perform well in practice with reasonably small sojourn periods. We designed two experiments, one representing a ride-sharing marketplace in Manhattan (Figure 3), and a freight marketplace operating in the 50 largest cities in the southeastern U.S. (Figure 4). We pair trip requests to reduce detour miles in the ride-sharing instance, and pair transportation requests to reduce empty driving miles in the freight marketplace instance.

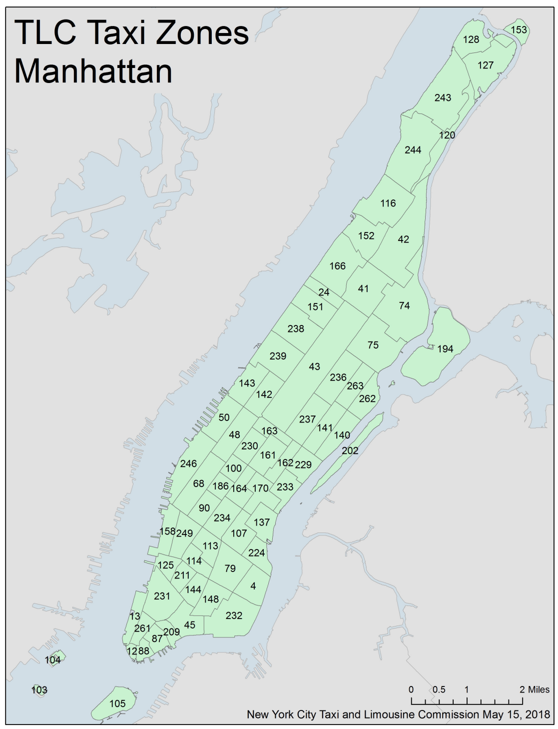

Figure 3: Taxi zones in Manhattan, New York City, accessed from NYC Taxi and Limousine Commission (2023)

6.1 Ride-Sharing

We design a ride-sharing instance based on the New York City Open Data platform, and following a similar instance construction to Lobel and Martin (2020). We construct a network of Manhattan's 69 taxi zones, and estimate the travel times between nodes in the network using OpenStreetMap data. We estimate the expected arrival rates of ride requests using the historical record of for-hire vehicles from New York City Open Data. The data includes the origin and destination taxi zones and pick-up and drop-off times for all the trips of

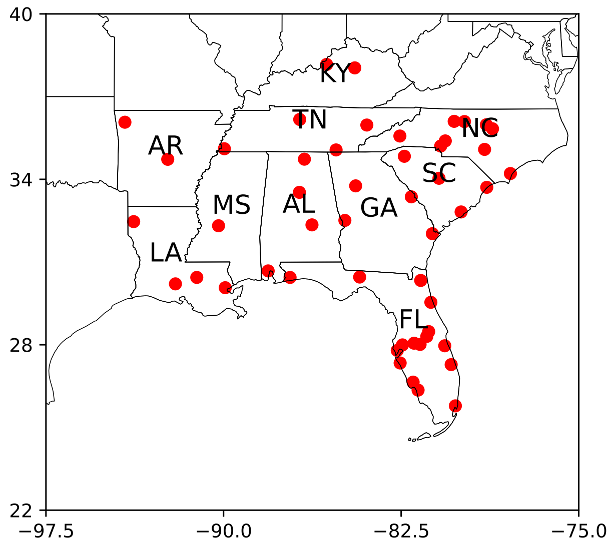

Figure 4: 50 largest cities in the southeastern US, used in freight marketplace experiment.

the major ride-sharing taxi platforms in New York City. For time homogeneity, we consider a one-month dataset (February 2020) during morning commute trips (6am to 10am, Monday to Friday except holidays), with origin and destination both located within Manhattan. We generate precise origins and destinations by sampling uniformly from within each taxi zone. Trip requests from an origin zone to a destination zone constitute a type, and we assume type arrivals follow independent Poisson distributions. The matching reward for a request pair is the travel distance saved following the optimal route among four possible routes, i.e. the distance saved if a driver performs two requests as a shared ride, calculated as the difference between the sum of distances required to fulfill both requests without ridesharing and the distance for a shared ride. There are four possible routes for a shared ride with two ride requests (O_i, D_i) and (O_j, D_j) (Lobel and Martin 2020):

1. $O_i \to O_j \to D_i \to D_j$
2. $O_i \to O_j \to D_j \to D_i$
3. $O_j \to O_i \to D_i \to D_j$
4. $O_j \to O_i \to D_j \to D_i$.

We assume the driver travels along the optimal route. Each trip request waits to be paired for the given sojourn period. If it reaches the end of the sojourn without being matched, we assume a driver serves it as a stand-alone ride; otherwise, a driver serves the two matched ride requests as a shared ride. Note that matching two nodes of the same type yields a positive reward in this case, which violates our assumption that $w_{ii} = 0$ for $i \in N$. However, same-type matches are extremely unlikely given the arrival rates implied by the data in our instance construction; we allowed same-type matches but did not observe a single one in any of our experiments.

For different sojourn periods, we compare the performance of four policies: myopic, greedy, batching, and offline. The myopic policy solves a maximum-reward matching every period and immediately executes the entire solution. Batching follows the procedures previously described. Greedy solves a maximum-reward matching problem every period, but only executes matches where the matched pair contains a node that is about to leave the system; it is a *delayed* greedy policy, which makes decisions as late as possible. Recall that a straightforward implementation of the delayed greedy policy may not be optimal, so in our theoretical analysis we modify it by slightly restricting the set of nodes the policy can match (cf. Section 4.2); however, our empirical results suggest that this modification is not usually necessary in practice. The offline solution solves the deterministic matching problem for the entire time horizon using realized values, and serves as an additional benchmark to the fluid relaxation. We set the time horizon length to one hour and collect requests and make matching decisions every 10 seconds. We test for five different sojourn periods of 60, 90, 120, 150, and 180 seconds. That is, we test for 6, 9, 12, 15, and 18 sojourn periods when the time horizon length is 360 periods. We simulate the horizon 200 times.

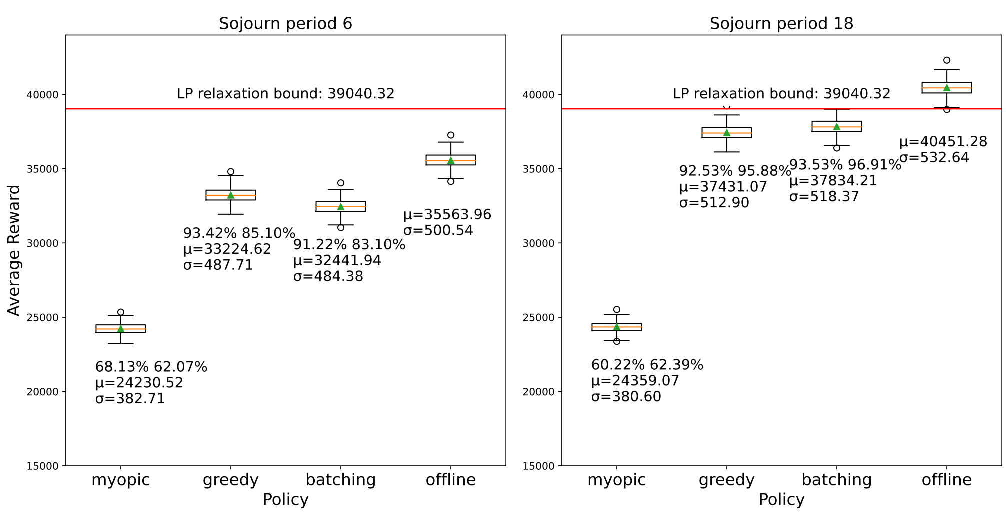

Figure 5: Empirical average rewards achieved by the tested policies in the ride-sharing instance.

Figure 5 shows the average reward achieved by each policy when the sojourn time is 60 and 180 seconds. The red line represents the LP relaxation bound, and μ and σ represent the empirical mean and standard deviation of each policy, respectively. The percentage values for each policy represent the ratio of the average reward obtained by the policy to the offline average value (left) and to the LP relaxation bound

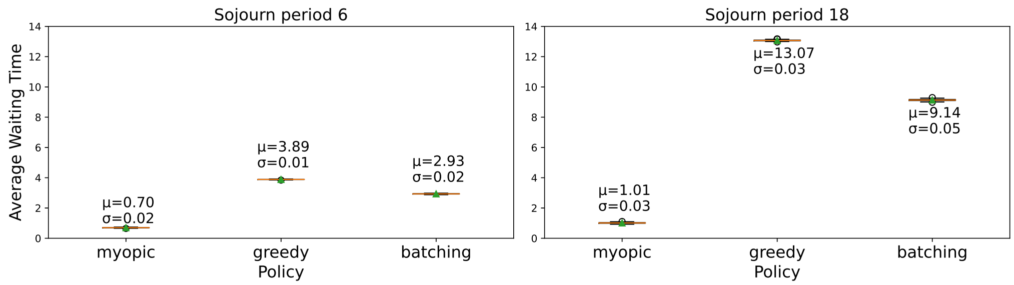

Figure 6: Empirical average waiting times achieved by the tested policies in the ride-sharing instance.

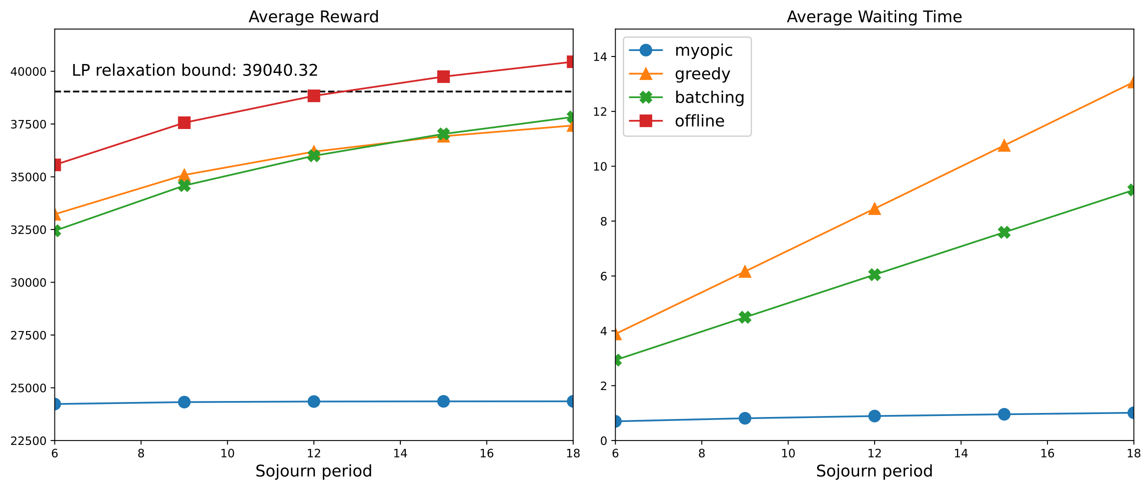

Figure 7: Empirical average rewards and waiting times achieved by the tested policies in the five different sojourn periods in the ride-sharing instance.

(right), respectively. We compute the LP relaxation bound by solving LP (3) with the expected rewards of taxi zone trip pairs instead of considering the precise origin and destination nodes in the network. The greedy and batching policies are both already within 90% of optimal for a sojourn period of 6, and achieve 95% or better with a sojourn period of 18; the myopic policy achieves less than 70% in all cases.

In our model, the rewards do not depend on when the corresponding matches are made and how long a node waits to be matched. Although the average waiting time is not considered in the objective function, it is an important secondary evaluation factor. Figure 6 shows the average waiting time achieved by each policy. As expected, the average waiting time increases from the myopic policy to the batching policy to the greedy policy; recall that the greedy policy waits until at least one node in a pair is about to expire before executing the match.

Figure 7 depicts the average reward and waiting time achieved by each policy in the five different sojourn period settings; we present more detailed results in Table 2 in Appendix B. As can be seen in Figure 7 and

Table 2, the average rewards of the greedy and batching policies get closer to the LP relaxation bound as the sojourn period increases, and the LP bound approaches and then becomes smaller than the offline reward starting at sojourn period 12. Somewhat surprisingly, the batching policy starts to beat the greedy policy at sojourn period 15. Note that the *delayed* greedy policy implemented in this experiment is slightly different from the *modified delayed greedy* policy proposed in Section 4.2. In addition, the average waiting time achieved by each policy appears close to linear with respect to the sojourn period in Figure 7. For the batching policy, agents wait in the system on average for half of the sojourn period, so the linear relationship is clear; the intuition is not as direct for the greedy policy, but is related to waiting until one node is about to expire to execute a match.

Since the performance of batching and greedy policies in terms of average reward is similar in this instance, a system manager might prefer to use the batching policy, because it makes arriving nodes wait significantly less on average.

6.2 Freight Marketplace

We estimate the number of daily requests in the marketplace using the statistics of freight flows by state in 2020, provided by the United States Department of Transportation (Bureau of Transportation Statistics 2023a); we use the average volume of freight flows per day with both origin and destination in Southeastern states. 64.6% of the volume is shipped by trucks on average (Bureau of Transportation Statistics 2023b), and the maximum capacity of a standard truck is 40 tons; this yields an estimate of the number of daily truckload shipment requests in the Southeastern region. Given an expected number of daily requests, we split these by state proportional to each state's average originating flow weight, then split the expected departures from state k into transport requests (k,ℓ) proportional to the average flow weight from state k to state ℓ. In each state k, we randomly choose a city k' with probability proportional to each city's population. Requests (k',ℓ') from city k' to ℓ' constitute a type. We assume daily requests for each type follow a Poisson distribution. The matching reward for a request pair is the deadhead distance saved, i.e. the distance saved if the same vehicle performs the requests sequentially instead of having each request served by a single vehicle that must deadhead back to the origin.

As in the experiment for the ride-sharing marketplace, we compare the performance of the same four policies. We test for sojourn periods of one through five days. We set the time horizon length to 200 days, and we simulate the horizon 200 times.

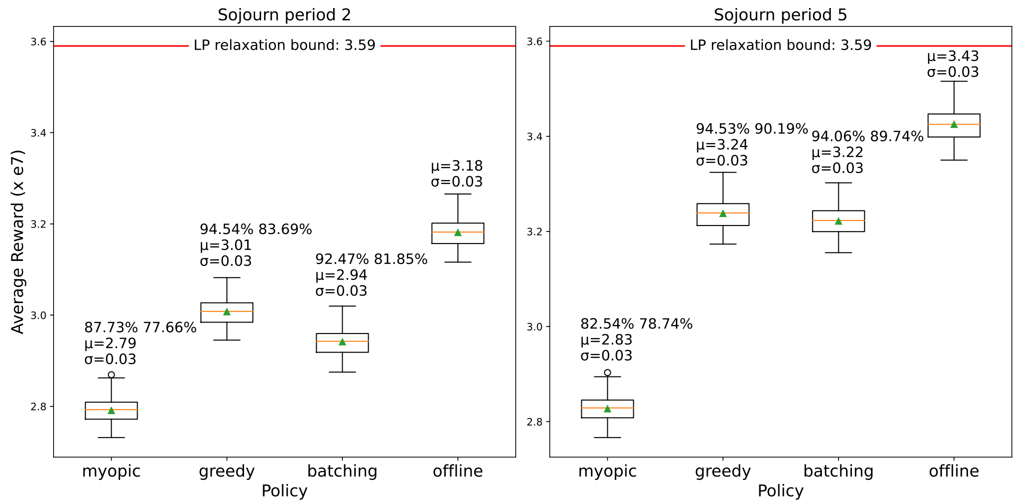

Figure 8: Average rewards achieved by the tested policies in the freight marketplace instance.

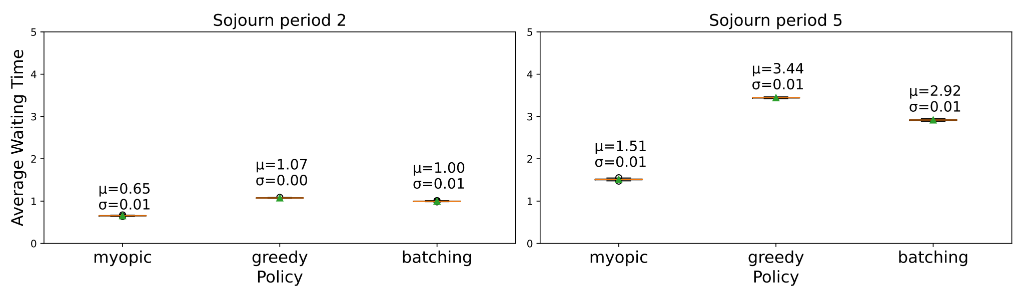

Figure 9: Average waiting times achieved by the tested policies in the freight marketplace instance.

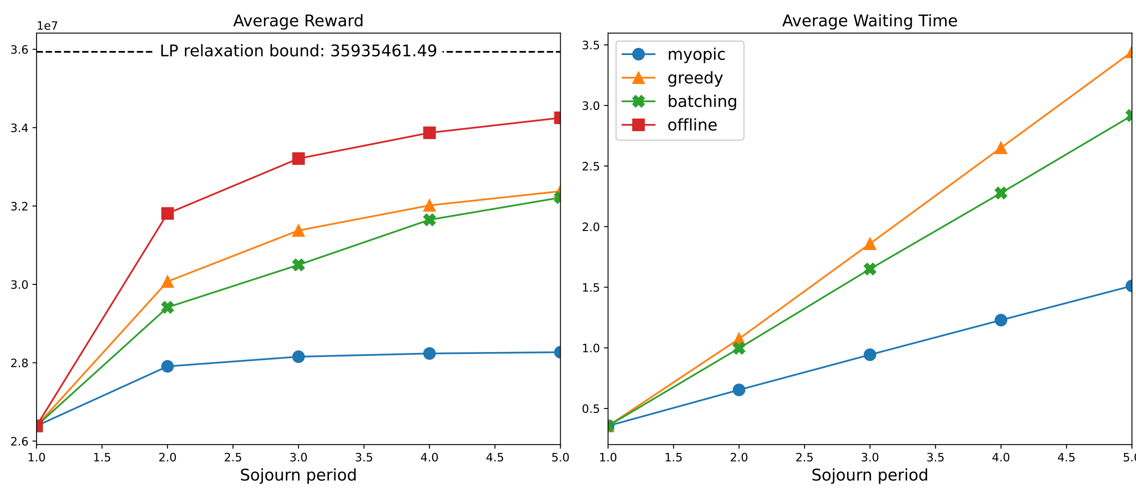

Figure 10: Average rewards and waiting times achieved by the tested policies for the five different sojourn periods in the freight marketplace instance.

Figures 8 and 9 respectively show the average reward and the average waiting time achieved by each policy when the sojourn time is two and five days. Figure 10 depicts the average reward and waiting time achieved by each policy in the five different sojourn period settings. The formats for Figures 8, 9, and 10 are the same as in Figures 5, 6, and 7, respectively. We present detailed results in Table 3 in Appendix B.

As can be seen in Figure 10 and Table 3, the average rewards of the greedy and batching policies get closer to the LP relaxation bound as the sojourn period increases. In addition, greedy converges to the LP relaxation bound faster than batching, although their performance is very close for the largest sojourn periods. Based on our results, if the system manager wants to provide a maximum waiting time guarantee of two days, they might prefer the greedy policy because it performs better than batching and there is no significant difference in average waiting times.

7 Conclusions

This work analyzes the asymptotic performance of two widely used policies, batching and greedy, for a dynamic, stochastic non-bipartite matching problem with a uniform sojourn time. The batching policy and a greedy policy with a randomized restriction on possible matches are both asymptotically optimal as the sojourn time grows; perhaps more importantly, both converge exponentially fast to approximate optimality. These results also extend to an impatient setting in which unmatched nodes leave the system with a certain probability, provided nodes are "patient enough." Our experiments show that the batching and greedy policies perform well in practice with reasonably small sojourn periods.

These results motivate studying a model in which a match's reward may depend on its nodes' waiting times, and an evaluation of how batching, greedy and other policies perform in this case. Similarly, our results also indicate that different management policies may be necessary in the presence of significant node impatience, which also motivates potential avenues for future research.

References

R. Anderson, I. Ashlagi, D. Gamarnik, and Y. Kanoria. Efficient dynamic barter exchange. *Operations Research*, 65 (6):1446–1459, 2017.

A. Aouad and Ö. Saritaç. Dynamic stochastic matching under limited time. *Operations Research*, 70(4):2349–2383, 2022.

I. Ashlagi, M. Burq, C. Dutta, P. Jaillet, A. Saberi, and C. Sholley. Edge weighted online windowed matching. *Mathematics of Operations Research*, 2022. (forthcoming).

A. Aveklouris, L. DeValve, and A. R. Ward. Matching impatient and heterogeneous demand and supply. *arXiv preprint arXiv:2102.02710*, 2021.

B. Bahmani and M. Kapralov. Improved bounds for online stochastic matching. In M. de Berg and U. Meyer, editors, *Algorithms – ESA 2010*, pages 170–181, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. ISBN 978-3-642-15775-2.

S. Behnezhad, J. Łącki, and V. Mirrokni. Fully dynamic matching: Beating 2-approximation in δ^ε update time. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2492–2508. SIAM, 2020.

A. Bernstein, A. Dudeja, and Z. Langley. A framework for dynamic matching in weighted graphs. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2021, page 668–681, New York, NY, USA, 2021. ACM. ISBN 9781450380539. doi: 10.1145/3406325.3451113. URL `https://doi.org/10.1145/3406325.3451113`.

S. Bhattacharya, M. Henzinger, and D. Nanongkai. New deterministic approximation algorithms for fully dynamic matching. In *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '16, page 398–411, New York, NY, USA, 2016. ACM. ISBN 9781450341325. doi: 10.1145/2897518.2897568. URL `https://doi.org/10.1145/2897518.2897568`.

S. Bhattacharya, P. Kiss, T. Saranurak, and D. Wajc. Dynamic matching with better-than-2 approximation in polylogarithmic update time. *arXiv preprint arXiv:2207.07438*, 2022.

J. Blanchet, M. Reiman, V. Shah, L. Wein, and L. Wu. Asymptotically optimal control of a centralized dynamic matching market with general utilities. *Operations Research*, 70, 01 2022. doi: 10.1287/opre.2021.2186.

B. Brubach, K. A. Sankararaman, A. Srinivasan, and P. Xu. New algorithms, better bounds, and a novel model for online stochastic matching. In *24th Annual European Symposium on Algorithms (ESA 2016)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

Bureau of Transportation Statistics. Freight flows by state. `https://www.bts.gov/browse-statistical-products-and-data/state-transportation-statistics/freight-flows-state`, 2023a. Accessed: 2023-06-21.

Bureau of Transportation Statistics. Freight shipments by mode. `https://www.bts.gov/topics/freight-transportation/freight-shipments-mode`, 2023b. Accessed: 2023-06-21.

N. Collina, N. Immorlica, K. Leyton-Brown, B. Lucier, and N. Newman. Dynamic weighted matching with heterogeneous arrival and departure rates. In *International Conference on Web and Internet Economics*, pages 17–30, Cham, 2020. Springer.

T. Ezra, M. Feldman, N. Gravin, and Z. G. Tang. Prophet matching with general arrivals. *Mathematics of Operations Research*, 47(2):878–898, 2022.

J. Feldman, A. Mehta, V. Mirrokni, and S. Muthukrishnan. Online stochastic matching: Beating 1-1/e. In *IEEE 50th Annual Symposium on Foundations of Computer Science (FOCS 2009)*, Los Alamitos, CA, USA, oct 2009. IEEE Computer Society. doi: 10.1109/FOCS.2009.72. URL `https://doi.ieeecomputersociety.org/10.1109/FOCS.2009.72`.

G. Feng, G. Kong, and Z. Wang. We are on the way: Analysis of on-demand ride-hailing systems. *Manufacturing & Service Operations Management*, 23(5):1237–1256, 2021.

V. Gupta. Greedy algorithm for multiway matching with bounded regret. *Operations Research*, 2022. (forthcoming).

P. Jaillet and X. Lu. Online stochastic matching: New algorithms with better bounds. *Mathematics of Operations Research*, 39(3):624–646, 2014.

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 '90, page 352–358, New York, NY, USA, 1990. ACM. ISBN 0897913612. doi: 10.1145/100216.100262. URL `https://doi.org/10.1145/100216.100262`.

S. Kerimov, I. Ashlagi, and I. Gurvich. On the optimality of greedy policies in dynamic matching. In *Proceedings of the 23rd ACM Conference on Economics and Computation*, EC '22, page 61, New York, NY, USA, 2022. ACM. ISBN 9781450391504. doi: 10.1145/3490486.3538323. URL `https://doi.org/10.1145/3490486.3538323`.

I. Lobel and S. Martin. Detours in Shared Rides. SSRN 3711072, 2020.

V. H. Manshadi, S. O. Gharan, and A. Saberi. Online stochastic matching: Online actions based on offline statistics. *Mathematics of Operations Research*, 37(4):559–573, 2012.

A. Mehta. Online matching and ad allocation. *Foundations and Trends in Theoretical Computer Science*, 8(4):265–368, 2013.

J. Montecinos, M. Ouhimmou, S. Chauhan, M. Paquet, and A. Gharbi. Transport carriers' cooperation on the last-mile delivery in urban areas. *Transportation*, 48(5):2401–2431, 2021.

M. Nazari and A. L. Stolyar. Reward maximization in general dynamic matching systems. *Queueing Systems*, 91(1): 143–170, 2019.

O. Neiman and S. Solomon. Simple deterministic algorithms for fully dynamic maximal matching. *ACM Transactions on Algorithms (TALG)*, 12(1):1–15, 2015.

NYC Taxi and Limousine Commission. Taxi zone map - manhattan. `https://www.nyc.gov/assets/tlc/images/content/pages/about/taxi_zone_map_manhattan.jpg`, 2023. Accessed: 2023-06-21.

E. Özkan and A. R. Ward. Dynamic matching for real-time ride sharing. *Stochastic Systems*, 10(1):29–70, 2020.

M. L. Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, USA, 2014.

A. Torrico and A. Toriello. Dynamic relaxations for online bipartite matching. *INFORMS Journal on Computing*, 34 (4):1871–1884, 2022.

A. Torrico, S. Ahmed, and A. Toriello. A polyhedral approach to online bipartite matching. *Mathematical Programming*, 172(1):443–465, 2018.

X. Wang, N. Agatz, and A. Erera. Stable matching for dynamic ride-sharing systems. *Transportation Science*, 52(4): 850–867, 2018.

C. Yan, H. Zhu, N. Korolko, and D. Woodard. Dynamic pricing and matching in ride-hailing platforms. *Naval Research Logistics (NRL)*, 67(8):705–724, 2020.

K. Zhang and Y. M. Nie. To pool or not to pool: Equilibrium, pricing and regulation. *Transportation Research Part B: Methodological*, 151:59–90, 2021.

A Appendix A: Additional Proofs

A.1 Proof of Proposition 3

Proof. We construct the fluid relaxation of (2) by introducing aggregating variables, $\alpha_e = \sum_{S\in\mathscr{S}} \sum_{x\in\mathscr{X}(S)} x_e z(S,x)$ for $e \in E$ and $\beta_{i,r} = \sum_{S\in\mathscr{S}} \sum_{x\in\mathscr{X}(S)} (S_i^r - \sum_{e\in\delta(i,r)} x_e) z(S,x)$ for $i \in N, r \in [\tau]$. The relaxation is

$$\max_{\alpha\geq 0,\beta\geq 0} \sum_{e\in E} w_e \alpha_e$$

$$\sum_{e\in\delta(i,\tau)} \alpha_e + \beta_{i,\tau} = \mathbb{E}[X_i], \quad i \in N \tag{4a}$$

$$\sum_{e\in\delta(i,r)} \alpha_e + \beta_{i,r} = \beta_{i,r+1}, \quad i \in N, r \in [\tau-1]. \tag{4b}$$

Variable α_e represents the average match rate of edge e and $\beta_{i,r}$ represents the average number of unmatched nodes of type i with r remaining periods. Constraints (4a) and (4b) are flow balance equations: each node in the system is matched with another node or remains unmatched. If it is unmatched, its remaining period decreases by 1.

Lemma 13. *The LP (4) is a relaxation of (2).*

We present its proof in Appendix A.2. We simplify the LP (4) by using its dual,

$$\min \sum_{i\in N} \mathbb{E}[X_i]\rho_{i,\tau}$$

$$\rho_{i,r} + \rho_{j,1} \geq w_{ij}, \quad i,j \in N, r \in [\tau] \tag{5a}$$

$$\rho_{i,r+1} \geq \rho_{i,r}, \quad i \in N, r \in [\tau-1] \tag{5b}$$

$$\rho_{i,1} \geq 0, \quad i \in N, \tag{5c}$$

where $\rho_{i,\tau}$ and $\rho_{i,r}$ are dual variables for constraint (4a) and (4b), respectively. Constraints (5a) are derived from the α variables, and constraints (5b) and (5c) from the β variables. Since the objective function depends only on the $\rho_{i,\tau}$ variables, we can rewrite (5) as

$$\min_{\rho\geq 0} \left\{ \sum_{i\in N} \mathbb{E}[X_i]\rho_{i,\tau} : \rho_{i,\tau} + \rho_{j,\tau} \geq w_{ij}, \quad i,j \in N \right\}, \tag{6}$$

and this LP's dual is

$$\max_{z\geq 0} \left\{ \sum_{i,j\in N} w_{ij} z_{ij} : \sum_{j\in N} z_{ij} \leq \mathbb{E}[X_i], \quad i \in N \right\}. \tag{3}$$

Thus, the LP (3) is equivalent to (4) and gives an upper bound on the original problem. □

A.2 Proof of Lemma 13

Proof. We first show that constraints (4a) are consequences of constraints (2a) and (2b). We can rewrite constraints (2a) as

$$\sum_{y\in\mathscr{X}(U^\tau,\dots,U^1)} z((U^\tau,\dots,U^1),y) = \phi(U^\tau)\sum_{S\in\mathscr{S}}\sum_{\substack{x\in\mathscr{X}(S)\\ :\,U^r=S^{r+1}(x)\ \forall r\in[\tau-1]}} z(S,x), \quad \forall\, U^\tau,\dots,U^1.$$

By summing over $U^{\tau-1},\dots,U^1$, we have for all U^τ,

$$\begin{aligned}\sum_{U^{\tau-1},\dots,U^1}\sum_{y\in\mathscr{X}(U^\tau,\dots,U^1)} z((U^\tau,\dots,U^1),y) &= \phi(U^\tau)\sum_{U^{\tau-1},\dots,U^1}\sum_{S\in\mathscr{S}}\sum_{\substack{x\in\mathscr{X}(S)\\ :\,U^r=S^{r+1}(x)\ \forall r\in[\tau-1]}} z(S,x)\\ &= \phi(U^\tau)\sum_{S\in\mathscr{S},x\in\mathscr{X}(S)} z(S,x) = \phi(U^\tau),\end{aligned}$$

where the last equality follows from constraints (2b). Recall that U^τ is a sequence $\{U_i^\tau\}_{i\in N}$. Fix type i and take summation over U^τ with weight U_i^τ. That is, for $i\in N$,

$$\sum_{U^\tau} U_i^\tau \sum_{U^{\tau-1},\dots,U^1}\sum_{y\in\mathscr{X}(U^\tau,\dots,U^1)} z((U^\tau,\dots,U^1),y) = \sum_{U^\tau} U_i^\tau\phi(U^\tau).$$

The left-hand-side is

$$\begin{aligned}&\sum_{U^\tau} U_i^\tau \sum_{U^{\tau-1},\dots,U^1}\sum_{y\in\mathscr{X}(U^\tau,\dots,U^1)} z((U^\tau,\dots,U^1),y)\\ &= \sum_{U\in\mathscr{S},y\in\mathscr{X}(U)} U_i^\tau z(U,y) = \sum_{U\in\mathscr{S},y\in\mathscr{X}(U)} \Big(\sum_{e\in\delta(i,\tau)} y_e + (U_i^\tau - \sum_{e\in\delta(i,\tau)} y_e)\Big) z(U,y) = \sum_{e\in\delta(i,\tau)}\alpha_e + \beta_{i,\tau},\end{aligned}$$

and the right-hand-side is

$$\sum_{U^\tau} U_i^\tau\phi(U^\tau) = \sum_{U_j^\tau\ \forall j\in N} U_i^\tau \prod_{j\in N}\phi(U_j^\tau) = \sum_{U_i^\tau} U_i^\tau \phi(U_i^\tau) = \mathbb{E}[X_i],$$

which yields constraints (4a). In a similar fashion, we show that constraints (4b) are consequences of constraints in model (2). Recall that $S_i^r = S_i^{r+1} - \sum_{e\in\delta(i,r+1)} x_e$ for $i\in N, r\in[\tau-1], x\in\mathscr{X}(S)$ given U because nodes with remaining period greater than 1 are matched only with expiring nodes. For $i\in N$ and $r\in[\tau-1]$,

$$\begin{aligned}\beta_{i,r+1} &= \sum_{S\in\mathscr{S},x\in\mathscr{X}(S)} (S_i^{r+1} - \sum_{e\in\delta(i,r+1)} x_e) z(S,x)\\ &= \sum_{S\in\mathscr{S},x\in\mathscr{X}(S)} S_i^r z(S,x)\\ &= \sum_{S\in\mathscr{S},x\in\mathscr{X}(S)} \Big(\sum_{e\in\delta(i,r)} x_e + (S_i^r - \sum_{e\in\delta(i,r)} x_e)\Big) z(S,x) = \sum_{e\in\delta(i,r)}\alpha_e + \beta_{i,r}. \qquad \square\end{aligned}$$

A.3 Omitted proof of Proposition 4

Proof. We prove that there exists an instance where the convergence rate is tight. Consider the case when X_i and X_j are Bernoulli random variables, both with probability $1/2$, in which case A_i and A_j are i.i.d. binomial; we show that $\min\{\mathbb{E}[X_i],\mathbb{E}[X_j]\} - \mathbb{E}[\min\{A_i,A_j\}]/\tau = \Omega(1/\sqrt{\tau})$. We first compute this value as follows:

$$\min\{\mathbb{E}[X_i],\mathbb{E}[X_j]\} - \frac{\mathbb{E}[\min\{A_i,A_j\}]}{\tau} = \frac{\mathbb{E}[A_i]}{\tau} - \frac{1}{\tau}\sum_{k=1}^{\tau}\mathbb{P}(\min\{A_i,A_j\}\geq k)$$

$$= \frac{1}{\tau}\sum_{k=1}^{\tau}\mathbb{P}(A_i\geq k) - \frac{1}{\tau}\sum_{k=1}^{\tau}\mathbb{P}(A_i\geq k)\mathbb{P}(A_j\geq k) = \frac{1}{\tau}\sum_{k=1}^{\tau}\mathbb{P}(A_i\geq k)\mathbb{P}(A_j\leq k-1)$$

$$= \frac{1}{\tau}\sum_{k=1}^{\tau}\sum_{\ell=k}^{\tau}\sum_{s=0}^{k-1}\left(\frac{1}{2}\right)^{2\tau}\binom{\tau}{\ell}\binom{\tau}{s} = \frac{1}{\tau}\left(\frac{1}{2}\right)^{2\tau}\sum_{\ell=1}^{\tau}\sum_{s=0}^{\ell-1}(\ell-s)\binom{\tau}{\ell}\binom{\tau}{s}$$

$$= \frac{1}{\tau}\left(\frac{1}{2}\right)^{2\tau}\sum_{k=1}^{2\tau-1}\sum_{\ell=1}^{\tau}\sum_{s=0}^{\ell-1}(\ell-s)\binom{\tau}{\ell}\binom{\tau}{s}1_{\{\ell+s=k\}}.$$

The fourth equality is from the binomial probability mass function, and the last two are rearranging terms. To simplify the right-hand-side of the last equality, we use the following lemma. Define $d(k) := \frac{1}{\tau}\sum_{\ell=1}^{\tau}\sum_{s=0}^{\ell-1}(\ell-s)\binom{\tau}{\ell}\binom{\tau}{s}1_{\{\ell+s=k\}}$ for $k\in[2\tau-1]$.

Lemma 14. $d(k) = \begin{cases}\binom{\tau-1}{\ell}^2 & \text{if } k=2\ell+1\\ \binom{\tau-1}{\ell-1}\binom{\tau-1}{\ell} & \text{if } k=2\ell.\end{cases}$

The proof for Lemma 14 is presented in Appendix A.4. Using this lemma, we get

$$\min\{\mathbb{E}[X_i],\mathbb{E}[X_j]\} - \frac{\mathbb{E}[\min\{A_i,A_j\}]}{\tau} = \sum_{k=1}^{2\tau-1}\left(\frac{1}{2}\right)^{2\tau}d(k) \geq \sum_{k=1,k\text{ is odd}}^{2\tau-1}\left(\frac{1}{2}\right)^{2\tau}d(k)$$

$$\overset{(a)}{=} \left(\frac{1}{2}\right)^{2\tau}\sum_{\ell=0}^{\tau-1}\binom{\tau-1}{\ell}^2 \overset{(b)}{=} \left(\frac{1}{2}\right)^{2\tau}\binom{2(\tau-1)}{\tau-1} \overset{(c)}{\geq} 4^{-\tau}\frac{4^{(\tau-1)}}{\sqrt{\pi(\tau-0.5)}} \geq \frac{1}{4\sqrt{\pi\tau}},$$

where equalities (a) and (b) follow from Lemma 14 and the Chu-Vandermonde identity, respectively, and (c) is from Stirling's formula. □

A.4 Proof of Lemma 14

Proof. It is known that $\ell\binom{\tau}{\ell} = \tau\binom{\tau-1}{\ell-1}$ and $\binom{\tau}{\ell} = \binom{\tau-1}{\ell} + \binom{\tau-1}{\ell-1}$. Plugging in it, we get

$$\begin{aligned}
d(k) &= \frac{1}{\tau}\sum_{\ell=1}^{\tau}\sum_{s=0}^{\ell-1}(\ell-s)\binom{\tau}{\ell}\binom{\tau}{s}1_{\{\ell+s=k\}} \\
&= \frac{1}{\tau}\sum_{\ell=1}^{\tau}\sum_{s=0}^{\ell-1}\left[\tau\binom{\tau-1}{\ell-1}\binom{\tau}{s} - \tau\binom{\tau}{\ell}\binom{\tau-1}{s-1}\right]1_{\{\ell+s=k\}} \\
&= \sum_{\ell=1}^{\tau}\sum_{s=0}^{\ell-1}\left[\binom{\tau-1}{\ell-1}\binom{\tau-1}{s} + \binom{\tau-1}{\ell-1}\binom{\tau-1}{s-1} - \binom{\tau-1}{\ell}\binom{\tau-1}{s-1} - \binom{\tau-1}{\ell-1}\binom{\tau-1}{s-1}\right]1_{\{\ell+s=k\}} \\
&= \sum_{\ell=1}^{\tau}\sum_{s=0}^{\ell-1}\left[\binom{\tau-1}{\ell-1}\binom{\tau-1}{s} - \binom{\tau-1}{\ell}\binom{\tau-1}{s-1}\right]1_{\{\ell+s=k\}}
\end{aligned}$$

Let $k = 2\ell+1$ for some ℓ. Then,

$$\begin{aligned}
d(k) = &\left[\binom{\tau-1}{2\ell}\binom{\tau-1}{0} - \binom{\tau-1}{2\ell+1}\binom{\tau-1}{-1}\right] \\
&+ \left[\binom{\tau-1}{2\ell-1}\binom{\tau-1}{1} - \binom{\tau-1}{2\ell}\binom{\tau-1}{0}\right] \\
&+ \left[\binom{\tau-1}{2\ell-2}\binom{\tau-1}{2} - \binom{\tau-1}{2\ell-1}\binom{\tau-1}{1}\right] + \dots \\
&+ \left[\binom{\tau-1}{\ell}\binom{\tau-1}{\ell} - \binom{\tau-1}{\ell+1}\binom{\tau-1}{\ell-1}\right] \\
= &\binom{\tau-1}{\ell}\binom{\tau-1}{\ell}.
\end{aligned}$$

In a similar way, we can verify the case for $k = 2\ell$. ☐

A.5 Omitted proof of Proposition 8

Proof. We need to prove

$$\Big[(1-p_i)(1-p_j) + (1-p_i)p_j\sum_{t\in[\tau]}(1-p_j)^{\tau-t}\left(\frac{p_i}{1-p_i}\right) + p_i(1-p_j)\sum_{t\in[\tau-1]}(1-p_i)^{\tau-t}(\frac{p_j}{1-p_j}) +$$
$$(1-p_i)(1-p_j)(1-p_j)^{\tau-1}\left(\frac{p_i}{1-p_i}\right) + (1-p_i)(1-p_j)(1-p_i)^{\tau-1}\left(\frac{p_j}{1-p_j}\right)\Big] = 1.$$

By canceling out and rearranging terms, we can rewrite the left-hand-side as

$$(1-p_i)(1-p_j) + p_ip_j\sum_{t\in[\tau]}(1-p_j)^{\tau-t} + p_ip_j\sum_{t\in[\tau-1]}(1-p_i)^{\tau-t} + p_i(1-p_j)^{\tau} + p_j(1-p_i)^{\tau}.$$

Using the geometric series, we get

$$
\begin{aligned}
&(1-p_i)(1-p_j)+p_ip_j\frac{1-(1-p_j)^\tau}{p_j}+p_ip_j\frac{1-p_i-(1-p_i)^\tau}{p_i}+p_i(1-p_j)^\tau+p_j(1-p_i)^\tau \\
=&(1-p_i)(1-p_j)+p_i-p_i(1-p_j)^\tau+p_j(1-p_i)-p_j(1-p_i)^\tau+p_i(1-p_j)^\tau+p_j(1-p_i)^\tau \\
=&(1-p_i)(1-p_j)+p_i+p_j(1-p_i)=1.
\end{aligned}
$$

□

A.6 Proof of Proposition 9

Proof. Let $S=(S_i,S_j)$ be an arbitrary state in $\mathscr{S}$. By Lemma 5, there are three possible cases:

1. $\mathscr{S}_i : T(S_i)\neq\emptyset, T(S_j)=\emptyset$
2. $\mathscr{S}_j : T(S_i)=\emptyset, T(S_j)\neq\emptyset$
3. $\mathscr{S}_0 : T(S_i)=\emptyset, T(S_j)=\emptyset$.

In case (1) (case (2)), the decision maker executes one match if $S_j^\tau=1$ ($S_i^\tau=1$). In case (3), the decision maker executes one match if $S_j^\tau=1$ and $S_i^\tau=1$. Thus, the long-run average matching frequency is

$$
\mathbb{P}(S_j^\tau=1)\sum_{S\in\mathscr{S}_i}\pi(S)+\mathbb{P}(S_i^\tau=1)\sum_{S\in\mathscr{S}_j}\pi(S)+\mathbb{P}(S_i^\tau=1)\mathbb{P}(S_j^\tau=1)\sum_{S\in\mathscr{S}_0}\pi(S),
$$

which is equivalent to

$$
\frac{1}{\sum_{S\in\mathscr{S}}h(S)}\Big[p_j\sum_{S\in\mathscr{S}_i}h(S)+p_i\sum_{S\in\mathscr{S}_j}h(S)+p_ip_j\sum_{S\in\mathscr{S}_0}h(S)\Big].
$$

We compute the sum of weights for states in $\mathscr{S}_i,\mathscr{S}_j$, and $\mathscr{S}_0$, respectively. *Case 1:* $p_i<p_j$. Consider $\sum_{S\in\mathscr{S}_i}h(S)$. For state $S\in\mathscr{S}_i$, $h(S)$ depends on the smallest remaining period $r=r(S_i)$ and the total number $m=|T(S_i)|$ of nodes of type i in the system. By calculating the summation with respect to r and m, we get

$$
\begin{aligned}
\sum_{S\in\mathscr{S}_i}h(S)&=\sum_{r\in[\tau-1]}\sum_{m\in[\tau-1]}\sum_{S\in\mathscr{S}_i}h(S)1_{\{\sum_{t\in[\tau-1]}S_i^t=m,\sum_{t\in[r-1]}S_i^t=0,S_i^r=1\}}\\
&=\sum_{r\in[\tau-1]}\sum_{m\in[\tau-1]}\sum_{S\in\mathscr{S}_i}(1-p_j)^{\tau-r}\Big(\frac{p_i}{1-p_i}\Big)^m1_{\{\sum_{t\in[\tau-1]}S_i^t=m,\sum_{t\in[r-1]}S_i^t=0,S_i^r=1\}}\\
&=\sum_{r\in[\tau-1]}\sum_{m\in[\tau-1]}(1-p_j)^{\tau-r}\Big(\frac{p_i}{1-p_i}\Big)^m\binom{\tau-1-r}{m-1}\\
&=\sum_{r\in[\tau-1]}(1-p_j)^{\tau-r}\sum_{m\in[\tau-r]}\Big(\frac{p_i}{1-p_i}\Big)^m\binom{\tau-1-r}{m-1}\\
&=\sum_{r\in[\tau-1]}(1-p_j)^{\tau-r}\Big(\frac{p_i}{1-p_i}\Big)\Big(1+\frac{p_i}{1-p_i}\Big)^{\tau-1-r}=\sum_{r\in[\tau-1]}\Big(\frac{1-p_j}{1-p_i}\Big)^{\tau-r}p_i=p_iq\frac{1-q^{\tau-1}}{1-q}.
\end{aligned}
$$

Similarly, we can calculate $\sum_{S\in\mathscr{S}_j} h(S)$ as

$$\sum_{S\in\mathscr{S}_j} h(S) = \frac{p_j}{q}\frac{1-q^{-\tau+1}}{1-q^{-1}}.$$

Using the definition of $\mathscr{S}_0$ and h, we can similarly calculate $\sum_{S\in\mathscr{S}_0} h(S)$ as

$$\sum_{S\in\mathscr{S}_0} h(S) = \sum_{S_i^\tau, S_j^\tau \in\{0,1\}} \mathbb{P}(X_i = S_i^\tau)\mathbb{P}(X_j = S_j^\tau) = 1.$$

Combining all of these results, we get the total sum of weights as

$$\sum_{S\in\mathscr{S}_i} h(S) + \sum_{S\in\mathscr{S}_j} h(S) + \sum_{S\in\mathscr{S}_0} h(S) = p_i q\frac{1-q^{\tau-1}}{1-q} + \frac{p_j}{q}\frac{1-q^{-\tau+1}}{1-q^{-1}} + 1,$$

and the long-run average matching frequency is

$$\begin{aligned}
&\frac{1}{p_i q\frac{1-q^{\tau-1}}{1-q} + p_j\frac{1}{q}\frac{1-q^{-\tau+1}}{1-q^{-1}} + 1}\left[p_j p_i q\frac{1-q^{\tau-1}}{1-q} + p_i p_j \frac{1}{q}\frac{1-q^{-\tau+1}}{1-q^{-1}} + p_i p_j\right] \\
&= p_i - \frac{p_i(1-p_j) - p_i(p_j - p_i)q\frac{1-q^{\tau-1}}{1-q}}{p_i q\frac{1-q^{\tau-1}}{1-q} + p_j\frac{1}{q}\frac{1-q^{-\tau+1}}{1-q^{-1}} + 1} \\
&= p_i - \frac{p_i(1-p_j) - p_i(1-p_j)(1-q^{\tau-1})}{p_i\frac{1-p_j}{p_j-p_i}(1-q^{\tau-1}) - p_j\frac{1-p_i}{p_j-p_i}(1-q^{-\tau+1}) + 1} \\
&= p_i - \frac{p_i(1-p_j)q^{\tau-1}}{\frac{p_j(1-p_i)q^{-\tau+1} - p_i(1-p_j)q^{\tau-1}}{p_j-p_i}} = p_i - \frac{(p_j-p_i)p_i(1-p_i)}{p_j(1-p_j)q^{-2\tau} - p_i(1-p_i)}.
\end{aligned}$$

Case 2: $p_i = p_j = p$. Using the same logic from Case 1, we obtain

$$\sum_{S\in\mathscr{S}_i} h(S) = (\tau-1)p, \sum_{S\in\mathscr{S}_j} h(S) = (\tau-1)p, \sum_{S\in\mathscr{S}_0} h(S) = 1, \sum_{S\in\mathscr{S}} h(S) = 2(\tau-1)p + 1.$$

The long-run average matching frequency is

$$\begin{aligned}
&\frac{1}{1+2(\tau-1)p}\left[p\sum_{S\in\mathscr{S}_i} h(S) + p\sum_{S\in\mathscr{S}_j} h(S) + p^2\sum_{S\in\mathscr{S}_0} h(S)\right] \\
&= \frac{1}{1+2(\tau-1)p}\left[p(\tau-1)p + p(\tau-1)p + p^2\right] = \frac{(2\tau-1)p^2}{1+2(\tau-1)p} = p - \frac{p(1-p)}{1+2(\tau-1)p}.
\end{aligned}$$

□

A.7 Omitted proof of Proposition 10

Proof. Fix $\ell \in [m]$. We need to show that $\sum_{S_k, k\in[m]\setminus\{\ell\}} \pi^c(S_1,\ldots,S_m) = \pi^B(S_\ell)$ for any S_ℓ. Define $\bar{\pi}(S_\ell) := \sum_{S_k, k\in[m]\setminus\{\ell\}} \pi^c(S_1,\ldots,S_m)$. We show that $\bar{\pi}$ is the stationary distribution of the Bernoulli chain. Fix U_ℓ, and let $P^c_{S,U}$ and $P^B_{S,U}$ be the transition probability from state S to U in the converted and Bernoulli chains, respectively, where S and U are defined corresponding to each chain:

$$
\begin{aligned}
\bar{\pi}(U_\ell) &= \sum_{U_k, k\in[m]\setminus\{\ell\}} \pi^c(U_1,\ldots,U_m) \\
&\overset{(a)}{=} \sum_{U_k, k\in[m]\setminus\{\ell\}} \sum_{(S_1,\ldots,S_m)} \pi^c(S_1,\ldots,S_m) P^c_{(S_1,\ldots,S_m),(U_1,\ldots,U_m)} \\
&\overset{(b)}{=} \sum_{U_k, k\in[m]\setminus\{\ell\}} \sum_{\substack{S_1: f(S_1)=g(U_1),\\ \ldots,\\ S_m: f(S_m)=g(U_m)}} \pi^c(S_1,\ldots,S_m) \mathbb{P}(X_{i,1}=U^\tau_{i,1}, X_{j,1}=U^\tau_{j,1},\ldots,X_{i,m}=U^\tau_{i,m}, X_{j,m}=U^\tau_{j,m}) \\
&= \sum_{U_k, k\in[m]\setminus\{\ell\}} \mathbb{P}(X_{i,1}=U^\tau_{i,1}, X_{j,1}=U^\tau_{j,1},\ldots,X_{i,m}=U^\tau_{i,m}, X_{j,m}=U^\tau_{j,m}) \sum_{\substack{S_1: f(S_1)=g(U_1),\\ \ldots,\\ S_m: f(S_m)=g(U_m)}} \pi^c(S_1,\ldots,S_m) \\
&= \sum_{U^\tau_k, g(U_k), k\in[m]\setminus\{\ell\}} \mathbb{P}(X_{i,1}=U^\tau_{i,1}, X_{j,1}=U^\tau_{j,1},\ldots,X_{i,m}=U^\tau_{i,m}, X_{j,m}=U^\tau_{j,m}) \sum_{\substack{S_1: f(S_1)=g(U_1),\\ \ldots,\\ S_m: f(S_m)=g(U_m)}} \pi^c(S_1,\ldots,S_m) \\
&= \sum_{g(U_k), k\in[m]\setminus\{\ell\}} \mathbb{P}(X_{i,\ell}=U^\tau_{i,\ell}, X_{j,\ell}=U^\tau_{j,\ell}) \sum_{\substack{S_1: f(S_1)=g(U_1),\\ \ldots,\\ S_m: f(S_m)=g(U_m)}} \pi^c(S_1,\ldots,S_m) \\
&= \mathbb{P}(X_{i,\ell}=U^\tau_{i,\ell}, X_{j,\ell}=U^\tau_{j,\ell}) \sum_{g(U_k), k\in[m]\setminus\{\ell\}} \sum_{\substack{S_1: f(S_1)=g(U_1),\\ \ldots,\\ S_m: f(S_m)=g(U_m)}} \pi^c(S_1,\ldots,S_m) \\
&= \mathbb{P}(X_{i,\ell}=U^\tau_{i,\ell}, X_{j,\ell}=U^\tau_{j,\ell}) \sum_{S_\ell: f(S_\ell)=g(U_\ell)} \sum_{\substack{g(U_k), k\in[m]\setminus\{\ell\}\\ S_k: f(S_k)=g(U_k)}} \pi^c(S_1,\ldots,S_m) \\
&= \mathbb{P}(X_{i,\ell}=U^\tau_{i,\ell}, X_{j,\ell}=U^\tau_{j,\ell}) \sum_{S_\ell: f(S_\ell)=g(U_\ell)} \sum_{S_k, k\in[m]\setminus\{\ell\}} \pi^c(S_1,\ldots,S_m) \\
&= \mathbb{P}(X_{i,\ell}=U^\tau_{i,\ell}, X_{j,\ell}=U^\tau_{j,\ell}) \sum_{S_\ell: f(S_\ell)=g(U_\ell)} \bar{\pi}(S_\ell) \overset{(c)}{=} \sum_{S_\ell} \bar{\pi}(S_\ell) P^B_{S_\ell,U_\ell}.
\end{aligned}
$$

The equality (a) follows because π^c is a stationary distribution. By rewriting the transition probability with functions f and g, we derive the equalities (b) and (c). Recall that $f(S)$ is the post-state of S and $g(U)$ is the sub-state of U except the newly arrived types. Other equalities are from rearranging the terms. □

A.8 Proof of Theorem 11

Proof. Let Y_i^t be the number of type i agents that remain in the system for t periods. With a slight abuse of notation, let A_i be the number of remaining type i arrivals within the batching interval τ, $A_i = \sum_{t=1}^{\tau-1} Y_i^t$. The long-run average matching frequency following the batching policy is $\mathbb{E}[\min\{A_i, A_j\}]/\tau$. In the proof of Proposition 4, we proved the following inequalities:

$$\mathbb{E}[\min\{A_i, A_j\}] \geq \mathbb{E}[A_i] - \frac{1}{2}\sqrt{\text{Var}[A_i] + \text{Var}[A_j]}, \tag{7}$$

$$\mathbb{E}[\min\{A_i, A_j\}] \geq \mathbb{E}[A_i] - 2\tau m \exp\Big(-\frac{2}{\tau m^2}(\mathbb{E}[A_j] - \mathbb{E}[A_i])^2\Big), \tag{8}$$

where m is the maximum number of type arrivals that can occur in a period.

To ease notation, we use d instead of $d_i(\tau)$. By the definition of Y_i^t, we can simply compute $\mathbb{E}[A_i]$ and get an upper bound of $\mathrm{Var}[A_i]$ as

$$\mathbb{E}[A_i] = \sum_{t=1}^{\tau}\mathbb{E}[Y_i^t] = \sum_{t=1}^{\tau}\sum_{n=1}^{m}\mathbb{P}(Y_i^t \geq n) = \sum_{t=1}^{\tau}\sum_{n=1}^{m}\mathbb{P}(X_i^t \geq n)(1-d)^{t-1} = \mathbb{E}[X_i]\frac{1-(1-d)^{\tau}}{d},$$

$$\begin{aligned}
\mathrm{Var}[A_i] &= \sum_{t=1}^{\tau}\mathrm{Var}[Y_i^t] = \sum_{t=1}^{\tau}\Big(\mathbb{E}[(Y_i^t)^2]-\mathbb{E}[Y_i^t]^2\Big) \leq \sum_{t=1}^{\tau}\mathbb{E}[(Y_i^t)^2] = \sum_{t=1}^{\tau}\sum_{n=1}^{m}n^2\mathbb{P}(Y_i^t = n)\\
&= \sum_{t=1}^{\tau}\sum_{n=1}^{m}n^2\sum_{n'=n}^{m}\mathbb{P}(X_i^t = n')\binom{n'}{n}(1-d)^{n(t-1)}(1-(1-d)^{t-1})^{n'-n}\\
&= \sum_{t=1}^{\tau}\sum_{n'=1}^{m}\mathbb{P}(X_i^t = n')\sum_{n=1}^{n'}n^2\binom{n'}{n}(1-d)^{n(t-1)}(1-(1-d)^{t-1})^{n'-n}\\
&= \sum_{t=1}^{\tau}\sum_{n'=1}^{m}\mathbb{P}(X_i^t = n')\Big(n'(1-d)^{t-1}+n'(n'-1)(1-d)^{2(t-1)}\Big)\\
&\leq \sum_{t=1}^{\tau}\mathbb{E}[X_i](1-d)^{t-1}+\sum_{t=1}^{\tau}\mathbb{E}[X_i^2](1-d)^{2(t-1)}\\
&= \mathbb{E}[X_i]\frac{1-(1-d)^{\tau}}{d}+\mathbb{E}[X_i^2]\frac{1-(1-d)^{2\tau}}{1-(1-d)^2}.
\end{aligned}$$

By plugging in $\mathbb{E}[A_i]$ and the upper bound of $\mathrm{Var}[A_i]$ to (7), we get

$$\frac{\mathbb{E}[\min\{A_i,A_j\}]}{\tau} \geq \mathbb{E}[X_i]\frac{1-(1-d)^{\tau}}{\tau d}-\frac{1}{2}\sqrt{(\mathbb{E}[X_i+X_j])\frac{1-(1-d)^{\tau}}{\tau^2 d}+\mathbb{E}[X_i^2+X_j^2]\frac{1-(1-d)^{2\tau}}{\tau^2(1-(1-d)^2)}}.$$

If $d=\mathscr{O}(\tau^{-\beta})$ where $\beta>1$, $\lim_{\tau\to\infty}\frac{1-(1-d)^{\tau}}{\tau d}=1$. Thus, the first term of the right-hand-side converges to $\mathbb{E}[X_i]$ as $\tau\to\infty$, and the second term is $\mathscr{O}(1/\sqrt{\tau})$. If $\mathbb{E}[X_i]<\mathbb{E}[X_j]$, we can use the stronger inequality (8), which results in

$$\frac{\mathbb{E}[\min\{A_i,A_j\}]}{\tau} \geq \mathbb{E}[X_i]\frac{1-(1-d)^{\tau}}{\tau d}-2m\exp\Big(-\frac{2}{\tau m^2}(\mathbb{E}[X_j]-\mathbb{E}[X_i])^2\Big(\frac{1-(1-d)^{\tau}}{d}\Big)^2\Big).$$

If $d=\mathscr{O}(\tau^{-\beta})$ where $\beta>1$, the first term of the right-hand-side converges to $\mathbb{E}[X_i]$ as $\tau\to\infty$, and the second term is $\mathscr{O}(e^{-\tau})$, because $\exp(\frac{2}{m^2}(\mathbb{E}[X_j]-\mathbb{E}[X_i])^2\tau)$ times the second term converges to a constant as $\tau\to\infty$.

□

A.9 Proof of Theorem 12

Proof. From Proposition 10, it suffices to show that the result is true for Bernoulli random arrivals. We assume node type arrivals follow Bernoulli distributions, $X_i \sim \mathrm{Bernoulli}(p_i), X_j \sim \mathrm{Bernoulli}(p_j)$ for $p_i, p_j \in (0,1]$. We claim that the following function satisfies the Markov chain's balance equations under the impatient node assumption. Define a function $h : \mathscr{S} \to \mathbb{R}$ with

$$h(S) = \mathbb{P}(X_i^\tau = S_i^\tau)\mathbb{P}(X_j^\tau = S_j^\tau) \times \begin{cases} (1-p_j)^{\tau - r(S_i)} \prod\limits_{t \in T(S_i)} \frac{(1-d_i(\tau))^{\tau-t} p_i}{1-(1-d_i(\tau))^{\tau-t} p_i} & \text{if } T(S_i) \neq \emptyset, T(S_j) = \emptyset \\ (1-p_i)^{\tau - r(S_j)} \prod\limits_{t \in T(S_j)} \frac{(1-d_j(\tau))^{\tau-t} p_j}{1-(1-d_j(\tau))^{\tau-t} p_j} & \text{if } T(S_i) = \emptyset, T(S_j) \neq \emptyset \\ 1 & \text{if } T(S_i) = \emptyset, T(S_j) = \emptyset, \end{cases} .$$

Recall $r(S_i) = \min\{t \in [\tau-1] : S_i^t = 1\}$ is the smallest remaining period of a node of type i in S, $T(S_i) = \{t \in [\tau-1] : S_i^t = 1\}$ is the set of periods where a node of type i exists in S, and $r(S_j)$ and $T(S_j)$ are defined analogously.

Let S be an arbitrary state in $\mathscr{S}$. Consider the subsequence of S, $g(S) = (g_i(S), g_j(S))$, as defined in the proof of Proposition 8. Recall that at least one of $g_i(S)$ and $g_j(S)$ is a zero sequence by Lemma 5. Without impatience, the transition probability from U to S is $\mathbb{P}(X_i^\tau = S_i^\tau)\mathbb{P}(X_j^\tau = S_j^\tau)$ if $f(U) = g(S)$, and zero otherwise. In contrast, when there are impatient nodes, the transition probability from U to S is

$$\mathbb{P}(X_i^\tau = S_i^\tau)\mathbb{P}(X_j^\tau = S_j^\tau) \times$$
$$\begin{cases} (1-d_i(\tau))^{|T(S_i)|} d_i(\tau)^{|L(S_i)|} & \text{if } T(S_i) \neq \emptyset, T(S_j) = \emptyset, f_i(U) = g_i(S) + \sum_{t \in L(S_i)} e_t, f_j(U) = g_j(S) \\ (1-d_j(\tau))^{|T(S_j)|} d_j(\tau)^{|L(S_j)|} & \text{if } T(S_i) = \emptyset, T(S_j) \neq \emptyset, f_i(U) = g_i(S), f_j(U) = g_j(S) + \sum_{t \in L(S_j)} e_t \\ d_i(\tau)^{|L(S_i)|} & \text{if } T(S_i) = \emptyset, T(S_j) = \emptyset, f_i(U) = g_i(S) + \sum_{t \in L(S_i)} e_t, f_j(U) = g_j(S) \\ d_j(\tau)^{|L(S_j)|} & \text{if } T(S_i) = \emptyset, T(S_j) = \emptyset, f_i(U) = g_i(S), f_j(U) = g_j(S) + \sum_{t \in L(S_j)} e_t \end{cases}$$

for some $L(S_i) \subseteq [\tau-1] \setminus T(S_i), L(S_j) \subseteq [\tau-1] \setminus T(S_j)$.

We first consider the case when $g_i(S)$ is not a zero sequence, i.e. $T(S_i) \neq \emptyset, T(S_j) = \emptyset$. To simplify the notation, we use d instead of $d_i(\tau)$ and denote $\frac{(1-d)^{\tau-t}}{1-(1-d)^{\tau-t} p_i}$ by $\alpha(t)$ for $t \in [\tau-1]$. It is straightforward to verify that $\alpha(t)(1 + d\alpha(t+1)) = (1-d)\alpha(t+1)$ for $t \in [\tau-2]$. We need to show that $\sum\limits_U h(U) P_{U,S} = \mathbb{P}(X_i^\tau - S_i^\tau)\mathbb{P}(X_j^\tau = S_j^\tau)(1-p_j)^{\tau - r(S_i)} \prod\limits_{t \in T(S_i)} \alpha(t)$. We classify the list of states U with $P_{U,S} > 0$ according to $r(U_i)$:

1. $\mathscr{U}_1 : r(U_i) \geq r(S_i) + 1$
2. $\mathscr{U}_2 : 2 \leq r(U_i) \leq r(S_i)$
3. $\mathscr{U}_3 : r(U_i) = 1$

Case (1) represents the situation when nodes of type i with remaining period t for $t \in L(S_i)$ leave the system being impatient. It occurs only when $S_j^\tau = 0$. Thus,

$$\sum_{U \in \mathscr{U}_1} h(U) P_{U,S} = (1-d)^{|T(S_i)|} \times \mathbb{P}(X_i = g_i^{\tau-1}(S)) \times \bar{h}(g_i'(S), 0) \times (1-p_j) \times \prod_{t \in L(S_i)} (1 + d\alpha(t+1)),$$

where $g_i'(S)$ is a subsequence of $g_i(S)$ excluding the first element, i.e. $g_i'(S) = (g_i^{\tau-2}(S), \ldots, g_i^1(S))$. The first term $(1-d)^{|T(S_i)|}$ represents the probability of nodes corresponding to $T(S_i)$ not leaving the system.

The last term $\prod_{t\in L(S_i)}(1+d\alpha(t+1))$ represents the sum of total probabilities having impatient nodes with remaining periods in $L(S_i)$. Note that an impatient node with remaining period t implies that its previous remaining period was $t+1$, which gives $\alpha(t+1)$. The other terms represent the transition from U to S, as described in the proof of Proposition 8.

Case (2) represents two situations. The first is when a node of type i with remaining period t for some $t < r(S_i)+1$ is matched with a newly arrived node of type j ($S_j^\tau = 1$) and other nodes of type i with remaining periods greater than t and in $\{t'+1 : t' \in L(S_i)\}$ abandon the system at the next period. The second situation is when $S_j^\tau = 0$ and nodes of type i with remaining periods in $\{t'+1 : t' \in L(S_i)\}$ abandon the system at the next period. Thus, we can derive

$$\sum_{U\in\mathscr{U}_2} h(U)P_{U,S} = (1-d)^{|T(S_i)|} \times \mathbb{P}(X_i = g_i^{\tau-1}(S)) \times \bar{h}(g_i'(S),0)\times \\ \sum_{t'=2}^{r(S_i)} (1-p_j)^{r(S_i)+1-t'} \times \alpha(t') \times \Big(p_j + d(1-p_j)\Big) \times \prod_{t\in L(S_i)\cup\{t',\dots,r(S_i)\}} (1+d\alpha(t+1)).$$

Case (3) is similar to case (2), but a node of type i with remaining period 1 leaves the system regardless of the existence of newly arrived node of type j. Thus, we have

$$\sum_{U\in\mathscr{U}_3} h(U)P_{U,S} = (1-d)^{|T(S_i)|} \times \mathbb{P}(X_i = g_i^{\tau-1}(S)) \times \bar{h}(g_i'(S),0)\times \\ (1-p_j)^{r(S_i)} \times \alpha(1) \times \prod_{t\in L(S_i)\cup\{1,\dots,r(S_i)\}} (1+d\alpha(t+1)).$$

Combining all the equalities, we get

$$\begin{aligned}
&\sum_U h(U)P_{U,S} \\
=&(1-d)^{|T(S_i)|}(1-p_j) \times \mathbb{P}(X_i = g_i^{\tau-1}(S)) \times \bar{h}(g_i'(S),0)\times \\
&\Big[\prod_{t\in L(S_i)}(1+d\alpha(t+1)) + \sum_{t'=2}^{r(S_i)} (1-p_j)^{r(S_i)+1-t'} \times \alpha(t') \times \Big(p_j + d(1-p_j)\Big) \times \prod_{t\in L(S_i)\cup\{t',\dots,r(S_i)\}} (1+d\alpha(t+1)) \\
&+ (1-p_j)^{r(S_i)} \times \alpha(1) \times \prod_{t\in L(S_i)\cup\{1,\dots,r(S_i)\}} (1+d\alpha(t+1))\Big] \\
=&(1-d)^{|T(S_i)|}(1-p_j) \times \mathbb{P}(X_i = g_i^{\tau-1}(S)) \times \bar{h}(g_i'(S),0) \times \prod_{t\in L(S_i)} (1+d\alpha(t+1)) \\
&\Big[1 + \alpha(S_i)\Big(p_j + d(1-p_j)\Big) \sum_{k=0}^{r(S_i)-2} \Big((1-p_j)(1-d)\Big)^k + (1-p_j)^{r(S_i)}(1-d)^{r(S_i)-1}\alpha(r(S_i))\Big] \\
=&(1-d)^{|T(S_i)|}(1-p_j) \times \mathbb{P}(X_i = g_i^{\tau-1}(S)) \times \bar{h}(g_i'(S),0) \times \prod_{t\in L(S_i)} (1+d\alpha(t+1)) \times (1+\alpha(r(S_i))) \\
=&(1-p_j)^{\tau-r(S_i)} \prod_{t\in T(S_i)} \alpha(t),
\end{aligned}$$

where the last equality follows from

$$
\begin{aligned}
&\mathbb{P}(X_i = g_i^{\tau-1}(S)) \times \bar{h}(g_i'(S), 0) \times \prod_{t\in L(S_i)} (1+d\alpha(t+1)) \\
=&\mathbb{P}(X_i = g_i^{\tau-1}(S))(1-p_j)^{\tau-r(S_i)-1} \prod_{t\in T(S_i)\setminus\{\tau-1\}} \alpha(t+1) \prod_{t\in L(S_i)} (1+d\alpha(t+1)) \\
=&\mathbb{P}(X_i = g_i^{\tau-1}(S))(1-p_j)^{\tau-r(S_i)-1} \prod_{t\in T(S_i)\setminus\{\tau-1\}} \alpha(t+1) \frac{\prod_{t\in\{r(S_i),\dots,\tau-1\}}(1+d\alpha(t+1))}{\prod_{t\in T(S_i)}(1+d\alpha(t+1))} \\
=&(1-p_j)^{\tau-r(S_i)-1} \frac{(1-d)^{\tau-r(S_i)}}{\alpha(r(S_i))} \prod_{t\in T(S_i)} \frac{\alpha(t+1)}{1+d\alpha(t+1)} \\
=&(1-p_j)^{\tau-r(S_i)-1} \frac{(1-d)^{\tau-r(S_i)-|T(S_i)|}}{\alpha(r(S_i))} \prod_{t\in T(S_i)} \alpha(t).
\end{aligned}
$$

We can verify the case when both $g_i(S)$ and $g_j(S)$ are zero sequences in an analogous way. We obtain the stationary distribution π by normalizing h, since $\mathscr{S}$ is finite. Using π, we compute the long-run average matching frequency of pair $\{i,j\}$ as

$$
\frac{p_j h(\mathscr{S}_i) + p_i h(\mathscr{S}_j) + p_i p_j}{h(\mathscr{S}_i) + h(\mathscr{S}_j) + 1} \tag{9}
$$

where $\mathscr{S}_i = \{S : \sum_{t\in[\tau-1]} S_i^t > 0, \sum_{t\in[\tau-1]} S_j^t = 0\}$ and $h(\mathscr{S}_i) = \sum_{S\in\mathscr{S}_i} h(S)$, and $\mathscr{S}_j$ and $h(\mathscr{S}_j)$ are defined analogously for node type j.

We next prove the long-run average matching frequency of pair $\{i,j\}$ in (9) converges to $\min\{p_i, p_j\}$ if $d_i(\tau) = d_j(\tau) = o(1)$. Without loss of generality, we assume that $p_i \le p_j$. We can rewrite (9) as

$$
\frac{p_j h(\mathscr{S}_i) + p_i h(\mathscr{S}_j) + p_i p_j}{h(\mathscr{S}_i) + h(\mathscr{S}_j) + 1} = p_i - \frac{p_i(1-p_j) - (p_j - p_i)h(\mathscr{S}_i)}{h(\mathscr{S}_i) + h(\mathscr{S}_j) + 1}, \tag{10}
$$

and it suffices to show the fractional term in (10) converges to 0 as $\tau \to \infty$. Following from the definition of function h, $\mathscr{S}_i$, and $\mathscr{S}_j$, we can obtain

$$
h(\mathscr{S}_i) = p_i \sum_{r=1}^{\tau-1} \Big((1-d_i(\tau))(1-p_j)\Big)^r \prod_{k=1}^{r} \frac{1}{1-(1-d_i(\tau))^k p_i},
$$

$$
h(\mathscr{S}_j) = p_j \sum_{r=1}^{\tau-1} \Big((1-d_j(\tau))(1-p_i)\Big)^r \prod_{k=1}^{r} \frac{1}{1-(1-d_j(\tau))^k p_j}.
$$

We consider the following two cases: (1) $p_i = p_j = p$, (2) $p_i < p_j$.

Case 1. $p_i = p_j = p$.

The fractional term in (10) is then $\frac{p(1-p)}{2h(\mathscr{S}_i)+1}$. Note that $h(\mathscr{S}_i) = h(\mathscr{S}_j)$ by symmetry. We show $h(\mathscr{S}_i)$ goes to ∞ as τ increases by using Fatou's lemma. Let $a(r,\tau) = \Big((1-d_i(\tau))(1-p)\Big)^r \prod_{k=1}^{r} \frac{1}{1-(1-d_i(\tau))^k p}$ for any positive integers r and τ. We can rewrite $h(\mathscr{S}_i)$ as

$$\frac{1}{p}h(\mathscr{S}_i)=\sum_{r=1}^{\tau-1}\Big((1-d_i(\tau))(1-p)\Big)^r\prod_{k=1}^{r}\frac{1}{1-(1-d_i(\tau))^k p}=\sum_{r=1}^{\tau-1}a(r,\tau)=\sum_{r=1}^{\infty}a(r,\tau)1_{\{r\leq\tau-1\}}.$$

By taking the limit inferior on both sides,

$$\begin{aligned}\liminf_{\tau\to\infty}\frac{1}{p_i}h(\mathscr{S}_i)&=\liminf_{\tau\to\infty}\sum_{r=1}^{\infty}a(r,\tau)1_{\{r\leq\tau-1\}}\\&\geq\sum_{r=1}^{\infty}\liminf_{\tau\to\infty}\Big(a(r,\tau)1_{\{r\leq\tau-1\}}\Big)\ \text{(by Fatou's lemma)}\\&\overset{\text{(a)}}{=}\sum_{r=1}^{\infty}1=\infty\end{aligned}$$

where equality (a) follows from

$$\begin{aligned}\liminf_{\tau\to\infty}\Big(a(r,\tau)1_{\{r\leq\tau-1\}}\Big)&=\liminf_{\tau\to\infty}a(r,\tau)\\&=\liminf_{\tau\to\infty}\Big((1-d_i(\tau))(1-p)\Big)^r\prod_{k=1}^{r}\frac{1}{1-(1-d_i(\tau))^k p}\\&=(1-p)^r\frac{1}{(1-p)^r}=1\end{aligned}$$

if $d_i(\tau)=o(1)$.

Case 2. $p_i<p_j$.

The fractional term in (10) is upper bounded by $\frac{p_i(1-p_j)}{h(\mathscr{S}_i)+h(\mathscr{S}_j)+1}$. We show $h(\mathscr{S}_j)$ goes to ∞ as τ increases by using Fatou's lemma again. Since $h(\mathscr{S}_j)$ is nonnegative, it then implies the fractional term in (10) converges to 0. Let $b(r,\tau)$ be $(1-d_j(\tau))(1-p_i)\Big)^r\prod_{k=1}^{r}\frac{1}{1-(1-d_j(\tau))^k p_j}$ for any positive integers r and τ. We can rewrite $h(\mathscr{S}_j)$ as

$$\frac{1}{p_j}h(\mathscr{S}_j)=\sum_{r=1}^{\tau-1}\Big((1-d_j(\tau))(1-p_i)\Big)^r\prod_{k=1}^{r}\frac{1}{1-(1-d_j(\tau))^k p_j}=\sum_{r=1}^{\tau-1}b(r,\tau)=\sum_{r=1}^{\infty}b(r,\tau)1_{\{r\leq\tau-1\}}.$$

By taking the limit inferior on both sides,

$$\begin{aligned}\liminf_{\tau\to\infty}\frac{1}{p_j}h(\mathscr{S}_j)&=\liminf_{\tau\to\infty}\sum_{r=1}^{\infty}b(r,\tau)1_{\{r\leq\tau-1\}}\\&\geq\sum_{r=1}^{\infty}\liminf_{\tau\to\infty}\Big(b(r,\tau)1_{\{r\leq\tau-1\}}\Big)\ \text{(by Fatou's lemma)}\\&\overset{\text{(b)}}{>}\sum_{r=1}^{\infty}1=\infty.\end{aligned}$$

where equality (b) follows from

$$\begin{aligned}\liminf_{\tau\to\infty}\left(b(r,\tau)1_{\{r\le\tau-1\}}\right) &= \liminf_{\tau\to\infty} b(r,\tau)\\ &= \liminf_{\tau\to\infty}\Big((1-d_j(\tau))(1-p_i)\Big)^r\prod_{k=1}^{r}\frac{1}{1-(1-d_j(\tau))^k p_j}\\ &= (1-p_i)^r\frac{1}{(1-p_j)^r} > 1\end{aligned}$$

if $d_j(\tau) = o(1)$.

We next show that if $d_i(\tau)$ is a constant for all $i \in N$, there exists an instance where the greedy policy does not converge to the optimal objective value of (3). Suppose $d_i(\tau) = 1-\varepsilon$ for some $\varepsilon > 0$ and $p_i = p_j = p$. Recall the definition of $h(\mathscr{S}_i)$,

$$\begin{aligned}\frac{1}{p}h(\mathscr{S}_i) &= \sum_{r=1}^{\tau-1}\Big((1-d_i(\tau))(1-p)\Big)^r\prod_{k=1}^{r}\frac{1}{1-(1-d_i(\tau))^k p} = \sum_{r=1}^{\tau-1}\Big(\varepsilon(1-p)\Big)^r\prod_{k=1}^{r}\frac{1}{1-\varepsilon^k p}\\ &\le \sum_{r=1}^{\tau-1}\Big(\varepsilon(1-p)\Big)^r\Big(\frac{1}{1-\varepsilon^\tau p}\Big)^r = \sum_{r=1}^{\tau-1}\Big(\frac{\varepsilon(1-p)}{1-\varepsilon^\tau p}\Big)^r = \frac{\varepsilon(1-p)}{1-\varepsilon^\tau p}\frac{1-\Big(\frac{\varepsilon(1-p)}{1-\varepsilon^\tau p}\Big)^\tau}{1-\Big(\frac{\varepsilon(1-p)}{1-\varepsilon^\tau p}\Big)}\\ &\le \frac{\varepsilon(1-p)}{1-\varepsilon^\tau p}\frac{1}{1-\Big(\frac{\varepsilon(1-p)}{1-\varepsilon^\tau p}\Big)}.\end{aligned}$$

By taking limits on both sides,

$$\lim_{\tau\to\infty}\frac{1}{p}h(\mathscr{S}_i) \le \lim_{\tau\to\infty}\frac{\varepsilon(1-p)}{1-\varepsilon^\tau p}\frac{1}{1-\Big(\frac{\varepsilon(1-p)}{1-\varepsilon^\tau p}\Big)} = \frac{\varepsilon(1-p)}{1-\varepsilon(1-p)},$$

which implies $h(\mathscr{S}_i)$ converges to a constant as τ increases. It follows that the long-run average matching frequency, which is the right-hand-side in (10), converges to a value smaller than p. □

B Appendix B: Experimental Study Results

Tables 2 and 3 show the exact average reward and waiting time values in the ride-sharing instance and the freight marketplace instance, respectively. The reward row consists of two sub-rows; the first row represents the 95% confidence interval of the average reward value, and the second represents the ratio of the average reward to the offline value and LP relaxation bound. The value written in the parentheses is the ratio to the LP relaxation bound.

Sojourn period	Result	Myopic	Greedy	Batching	Offline
6	Reward	24230.52 ± 53.50	33224.62 ± 68.18	32441.94 ± 67.71	35563.96 ± 69.97
		68.13% (62.07%)	93.42% (85.10%)	91.22% (83.10%)	-
	Waiting time	0.70	3.89	2.93	-
9	Reward	24322.27 ± 53.15	35086.75 ± 68.59	34583.92 ± 69.37	37563.31 ± 72.05
		64.75% (62.30%)	93.41% (89.87%)	92.07% (88.59%)	-
	Waiting time	0.81	6.16	4.49	-
12	Reward	24349.39 ± 53.70	36191.17 ± 69.31	35995.65 ± 70.80	38835.83 ± 73.09
		62.70% (62.37%)	93.19% (92.70%)	92.69% (92.20%)	-
	Waiting time	0.89	8.45	6.05	-
15	Reward	24357.53 ± 53.25	36922.11 ± 71.46	37030.52 ± 71.73	39746.28 ± 73.99
		61.28% (62.39%)	92.89% (94.57%)	93.17% (94.85%)	-
	Waiting time	0.96	10.76	7.59	-
18	Reward	24359.07 ± 53.21	37431.07 ± 71.70	37834.21 ± 72.46	40451.28 ± 74.46
		60.22% (62.39%)	92.53% (95.88%)	93.53% (96.91%)	-
	Waiting time	1.01	13.07	9.14	-

Table 2: Empirical average rewards and waiting times achieved by the tested policies in the ride-sharing instance.

Sojourn period	Result	Myopic	Greedy	Batching	Offline
1	Reward ($\times 10^7$)	2.64 ± 0.0039	2.64 ± 0.0039	2.64 ± 0.0039	2.64 ± 0.0039
		73.43% (100%)	73.43% (100%)	73.43% (100%)	-
	Waiting time	0.36	0.36	0.36	-
2	Reward ($\times 10^7$)	2.79 ± 0.0039	3.01 ± 0.0042	2.94 ± 0.0041	3.18 ± 0.0043
		77.66% (87.73%)	83.69% (94.54%)	81.85% (92.47%)	-
	Waiting time	0.65	1.07	1.00	-
3	Reward ($\times 10^7$)	2.82 ± 0.0039	3.14 ± 0.0044	3.05 ± 0.0043	3.32 ± 0.0045
		78.58% (83.37%)	89.10% (94.53%)	88.07% (93.44%)	-
	Waiting time	0.94	1.86	1.65	-
4	Reward ($\times 10^7$)	2.83 ± 0.0039	3.20 ± 0.0045	3.16 ± 0.0045	3.39 ± 0.0047
		78.67% (82.54%)	90.10% (94.53%)	89.65% (94.06%)	-
	Waiting time	1.23	2.65	2.28	-
5	Reward ($\times 10^7$)	2.83 ± 0.0039	3.24 ± 0.0045	3.22 ± 0.0045	3.43 ± 0.0048
		78.67% (82.54%)	90.10% (94.53%)	89.65% (94.06%)	-
	Waiting time	1.51	3.44	2.92	-

Table 3: Empirical average rewards and waiting times achieved by the tested policies in the freight marketplace instance.