

Decentralized Matching in a Probabilistic Environment

MOBIN Y. JELOUDAR, Stanford University

IRENE LO, Stanford University

TRISTAN POLLNER, Stanford University

AMIN SABERI, Stanford University

ABSTRACT

We consider a model for repeated stochastic matching where compatibility is probabilistic, is realized the first time agents are matched, and persists in the future. Such a model has applications in the gig economy, kidney exchange, and mentorship matching.

We ask whether a *decentralized* matching process can approximate the optimal online algorithm. In particular, we consider a decentralized *stable matching* process where agents match with the most compatible partner who does not prefer matching with someone else, and known compatible pairs continue matching in all future rounds. We demonstrate that the above process provides a 0.316-approximation to the optimal online algorithm for matching on general graphs. We also provide a $1/7$ -approximation for many-to-one bipartite matching, a $1/11$ -approximation for capacitated matching on general graphs, and a $1/2k$ -approximation for forming teams of up to k agents. Our results rely on a novel coupling argument that decomposes the successful edges of the optimal online algorithm in terms of their round-by-round comparison with stable matching.

CCS Concepts: • Mathematics of computing → Matchings and factors; Approximation algorithms; • Theory of computation → Algorithmic mechanism design; Stochastic approximation; • Social and professional topics → Centralization / decentralization.

Additional Key Words and Phrases: Matching Theory, Stochastic Matching, Approximation Algorithms, Decentralized Algorithms, Stable Matching, Matching Hypergraphs

ACM Reference Format:

Mobin Y. Jeloudar, Irene Lo, Tristan Pollner, and Amin Saberi. 2021. Decentralized Matching in a Probabilistic Environment. In *Proceedings of the 22nd ACM Conference on Economics and Computation (EC '21), July 18–23, 2021, Budapest, Hungary*. ACM, New York, NY, USA, 19 pages. <https://doi.org/10.1145/3465456.3467652>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

EC '21, July 18–23, 2021, Budapest, Hungary

© 2021 Association for Computing Machinery.

ACM ISBN 978-1-4503-8554-1/21/07...\$15.00

<https://doi.org/10.1145/3465456.3467652>

1 INTRODUCTION

We consider a model with a finite set of agents who can be repeatedly matched in each of a finite number of rounds. Each pair of agents (i, j) is compatible with a known probability $p_{\{i,j\}}$. When a pair is first matched, their compatibility is realized, and is successful with probability $p_{\{i,j\}}$ and unsuccessful with probability $1 - p_{\{i,j\}}$. This compatibility persists through all future rounds. This model captures learning dynamics in platforms that match workers with repeated tasks (dog-walking, babysitting, private chefs), mentorship programs, and kidney exchange. In such models, agents may only have an estimate of their compatibility with potential partners and typically learn their compatibility with others through being matched.

The goal of a matching platform is to maximize a *weighted sum of the size of the matching in each round*. For example, a platform may want to maximize the total number of successful matches, such as the total number of dog-dogwalker days or babysitting events. Or, it may only be interested in the number of successfully matched pairs by the end of the matching process. In mentorship and kidney exchange programs agents delays are costly, so matches made earlier in the process may be more valuable than matches made later. In contrast, the goal of each agent is to selfishly maximize the number of rounds in which they are matched.

Consider two matching processes. The optimal centralized matching algorithm, OPT, solves an NP-hard Bayesian optimization problem to maximize the expected reward, by prescribing matchings adaptively across rounds. In contrast, in the Stable Matching or SM process, the pairs that were successfully matched to each other in previous rounds remain matched and the rest of the matching is formed by pairing the agents with the highest success probability in a greedy fashion. Ties are broken arbitrarily. We show that SM provides a constant-factor approximation of OPT.

THEOREM 1. *The expected reward of SM is at least a 0.316-approximation of the expected reward of the optimal online algorithm.*

In Section 2, we provide a more detailed explanation of the SM process and its relationship with stable matching. Informally, if agents form preference lists over all other agents based on probability of compatibility, then SM forms a stable matching with respect to these preferences.

Stable matching has a number of attractive properties. While the optimal centralized matching OPT is NP-hard to compute, stable matching can be computed in polynomial time and reached in a decentralized manner. Additionally, also in contrast with OPT, stable matching is incentive compatible: it does not match agents against their will by forcing unmatched agents with high compatibility to match with low-compatibility partners, or break up compatible matches in future rounds. Stable matching is also attractive in settings such as kidney exchange or mentor matching where time is valuable, as it does not sacrifice present potential matches for future payoffs.

Our analysis implies a better approximation factor for another previously-known matching algorithm, known as GREEDY-COMMIT[9]. Similar to SM, GREEDY-COMMIT commits to known compatible pairs by matching them in all future rounds, but unlike SM, it selects a maximum weight matching between the remaining agents to maximize the immediate expected reward. We can interpret GREEDY-COMMIT as an off-the-shelf centralized matching service that can be computed by the platform in polynomial time if it knows the compatibility probabilities but does not want to solve the optimal online repeated matching problem. We show that GREEDY-COMMIT provides a better constant-factor approximation to OPT.

THEOREM 2. *The expected reward of GREEDY-COMMIT is at least a 0.43-approximation of the expected reward of the optimal online algorithm.*

Finally, we consider settings where agents have the capacity to be bilaterally matched with multiple other agents, or grouped in teams. Many gig economy markets and matching settings are

built on many-to-one bipartite markets, where agents on one side (e.g. workers) can be matched to multiple agents on the other side of the bipartition (e.g. tasks). In this setting SM provides a $1/7$ -approximation to OPT. In the general matching problem where all agents can be matched to multiple other agents, SM provides a $1/11$ -approximation to OPT, and in the teams problem where we form teams of up to k agents, SM provides a $1/2k$ -approximation to OPT.

Together, our results contribute to the discussion of the benefits of investing in optimal centralized matching. We show if the platform allows myopic, self-interested agents to determine a decentralized matching, the outcome attains a constant factor of the expected reward of the optimal online matching. If the platform offers a centralized matching service, in addition to determining match probabilities it must also force agents into matches that are not incentive-compatible, and may even need to solve an NP-hard problem.

1.1 Technical Ingredients

A key technical contribution of this paper is to compare the decentralized stable matching process to the optimal *online* algorithm for centralized repeated matching. In the study of online Bayesian optimization, an extremely common benchmark is the optimal *offline* algorithm, which knows at the outset which edges are successful and unsuccessful. In our setting, optimal offline is not an interesting benchmark, because for a certain family of inputs any online algorithm achieves an arbitrarily poor approximation to the optimal offline algorithm (see Appendix A).

We compare the stable matching SM with the optimal online algorithm OPT by *coupling* edges selected by OPT with those selected by SM. Since the two algorithms uncover *different information* as they progress, care must be taken in the coupling process not to condition analysis of the reward achieved by one algorithm on information that is acquired by the other.

Domination Lemma. A pivotal piece of our analysis is a *Domination Lemma*, which uses the greedy structure of the stable matching to bound the reward generated by a subset of edges selected by OPT by twice the reward generated by SM. The subset of edges is carefully chosen so that SM approximately improves upon OPT, despite the fact that OPT is able to use information from prior rounds in a more sophisticated manner. Specifically, note that SM greedily selects max-weight edges between agents who have not yet been matched. Now fix a round t , and consider the edges selected by OPT in round t that do not share an endpoint with successful edges previously selected (i.e. selected in rounds $1, 2, \dots, t-1$) by SM. Such edges can also be selected by SM in round t , and so the greediness of SM implies that the expected number of such edges that are successful is bounded by the twice the expected number of successful edges newly selected by SM in round t . While the intuition behind the Domination Lemma is straightforward, care needs to be taken to ensure that expectations are taken appropriately, since OPT and SM explore different parts of the sample space and so their performance is conditional on different histories.

Charging Lemma. We bound the remaining edges using a *Charging Lemma* that charges edges selected by OPT to adjacent edges selected by SM. For matching on general graphs, the bounds provided by the Charging Lemma and a refined version of the Domination Lemma define a factor-revealing LP that yields the 0.316-approximation. In the other settings, generalized versions of the Charging Lemma and Domination Lemma show the constant-factor approximations.

Our analysis also provides some intuition for why stable matching, which utilizes a myopic, decentralized matching process in each round, and which limits its use of information by committing to successful matches, is nonetheless able to achieve a constant-factor approximation to the optimal online algorithm, which can select optimal matchings in each round, and can also adaptively make use of information across rounds. In the uncapacitated matching setting, the Domination

Lemma shows that in any given round, agents who have not previously been matched can match themselves via stable matching at least half as well as OPT, and the Charging Lemma observes that in any given round, agents who are matched in that round via stable matching are collectively matched at least half as well compared to OPT. Therefore stable matching cannot do much worse than OPT, even though it makes use of information in a much less sophisticated manner.

1.2 Relationship with Prior Work

There is a large body of work on *centralized* and *decentralized* matchings in online and two-sided platforms. Most of this literature focuses on designing optimal information structures in centralized matchings, or optimal procedures for centralized matching by the platform. Our question also motivates a variant of *repeated stochastic matching* that is related to existing literature on the *query-commit problem* and *stochastic matching with rewards*. Interestingly, our analysis is sufficiently general to also handle previously studied centralized algorithms in the query-commit setting, and consequently we are able to provide improved bounds on prior algorithms.

Optimal matching in matching platforms. There is also a substantial literature on finding optimal or near-optimal matching policies in two-sided platforms with long-lived agents, such as matching in ridesharing [5, 6, 22, 23], volunteer platforms [33] and blood donation [35]. We focus on such platforms where match compatibility is difficult to determine and can be learned exactly only through matching. A well-known example of such a setting is kidney exchange, which has been studied from a repeated matching perspective [3, 4] and a failure-aware perspective [11].

Online and Stochastic Matching. The online bipartite matching problem introduced in [28], where vertices on one side of a bipartite graph arrive online, is foundational to the literature on online matching problems. Many variations have been studied, including the *adwords problem* [16, 37], *matching with stochastic rewards* [18, 24, 36, 38] (where edge realizations are stochastic) and the random-order and i.i.d. online matching problems [10, 12, 25, 27, 29, 30, 32, 34]. While almost all of these papers provide competitive guarantees compared to the optimum offline algorithm, we provide guarantees compared to the optimum online algorithm in a multi-round environment.

Query-Commit. Motivated in part by the application of kidney exchange, there is a large body of literature studying a variant of stochastic matching known as the query-commit problem [1, 7, 9, 14, 17, 21, 40]. Here, edges have fixed realization probabilities and can be queried in sequence, with the constraint that successful edges queried must be accepted, and accepted edges must form a matching. The objective is to maximize the size of the resulting matching. [9] introduced this setting and proved that the greedy algorithm (which simply queries feasible edges in decreasing order of probability) is a $1/4$ -approximation to the optimal online algorithm even when vertices have *patience parameters*, i.e. limits on how many incident edges can be queried. The authors additionally showed that the optimal online algorithm is NP-hard to compute. [1] improved on this result by showing that the greedy algorithm is in fact a $1/2$ -approximation (with patience parameters), and [7] provided constant-factor approximations on weighted graphs through an LP-based approach. Later work provided constant-factor approximations in query-commit settings for feasibility constraints significantly generalizing the matching constraint (e.g., [19], [20], [2]).

A less common assumption in the query-commit literature is the ability to query a matching in each round, instead of a single edge. [9] define this setting, propose the GREEDY-COMMIT algorithm, and prove that it achieves a $1/4$ -approximation to the optimal online algorithm which is forced to commit. Their analysis can furthermore be extended to a *capacitated matching* setting, where in each round we can select a matching that must respect integral capacity constraints on each vertex. [7] also prove a $1/20$ -approximation in the weighted setting, where the constraint is that a matching of size at most C can be queried in each round (for any parameter C).

The main differences between this line of work and our contribution are two-fold. Most importantly, while GREEDY-COMMIT operates *globally* in each round, the decentralized matching process we analyze cannot replicate such a centralized greedy approach. Secondly, we compare to algorithms that are not forced to commit to successful edges. (If OPT-Commit denotes the optimal online algorithm that is forced to forever match any successful edges it finds, simple examples show that OPT is a more powerful algorithm than OPT-Commit; see Appendix C.) Because we compare to the optimal online algorithm that is not restricted to committing, our coupling argument is substantively different to, and more delicate than, the one in [9]. Despite this challenge, our analysis of the decentralized setting still improves upon [9]; we show Greedy achieves a 0.316-approximation against a non-committing (more powerful) OPT, tightening the previous analysis of a $1/4$ -approximation to committing OPT.

Learning by Matching. Prior work has also explored several models of learning in matching settings. Many of these papers consider a multi-armed bandit setting where rewards are stochastic and redrawn i.i.d. from an unknown distribution, with algorithms of an explore-exploit nature [13, 26, 31]. Efficient algorithms that approximate the optimal online algorithm were also studied by [41] in a stochastic setting with a price for querying each edge, relying on a Gittins Index characterization for the optimal online algorithm [15, 42].

2 STABLE MATCHING IN GENERAL GRAPHS

We begin with a setting where each agent is matched with at most one other agent in each round. This setting captures bipartite matching problems such as matching mentors and mentees, as well as matching in general graphs such as matching peer mentors, roommates and kidney exchange.

There is a set V of vertices, representing agents. For every pair $i, j \in V$ of agents with $i \neq j$, there is an edge between i and j with probability $p_{\{i,j\}}$ independent from other edges, representing the compatibility of the agents. The set of agents and probabilities are known at the outset. We will find it useful to view the entire graph as being generated randomly from the outset: before the first round nature samples the graph $G(V, E)$ with probability $\prod_{e \in E} p_e \prod_{e \notin E} (1 - p_e)$. The platform and agents do not have direct access to G . Instead, in rounds $1 \leq t \leq T$, they can determine a matching M_t between the vertices in V and observe whether the edges in M_t are in E or not.

Let X_e be the Bernoulli random variable indicating whether e is in E ; we say an edge is *successful* if $X_e = 1$. Each agent's goal is to maximize the expected number of rounds in which they are successfully matched. The platform's reward in each round is equal to the number of successful edges selected in that round. Given weights $\omega_1, \dots, \omega_T$, the platform's goal is to maximize the weighted sum $\sum_{t=1}^T \omega_t \sum_{e \in M_t} X_e$ of the rewards collected in all T rounds. This can capture if the platform's goal is to maximize the size of the matching in the last round ($\omega_t = \mathbb{1}_{t=T}$), maximize the total number of successful matches in each round ($\omega_t = 1$ for all t), or weighted toward favoring matches in earlier rounds (e.g. $\omega_t = \delta^t$ for $\delta \in (0, 1)$).

The (deterministic) optimal online algorithm for maximizing the total reward can be derived by an exponential-size dynamic program [8]. However, exact optimization is NP-hard (as we discuss later), and for large numbers of agents it is hence infeasible for a matching platform to compute this algorithm. With this in mind, we focus on simple matching processes that are computable in polynomial time, and give approximations to the optimal online algorithm.

We assume that each agent i knows his or her compatibility probabilities $p_{\{i,j\}}$ with every other agent j . This could be enabled, for example, by search functionality in the platform that allows agents to view information about other agents. Fix a given round and an agent i , and assume that through prior matches agent i now has *updated* their priors with other agents to $\{\hat{p}_{\{i,j\}}\}$ (i.e., all unsuccessful matches are updated to 0 and all successful matches are updated to 1). Agent i wants

to be matched as soon as possible, so in this round would like to match with the remaining agent $\operatorname{argmax}_{j \neq i} \hat{p}_{\{i,j\}}$ who is most likely to be compatible with i . Hence, agent i can form a preference list over all other agents in this round by sorting their current compatibility probabilities $\{\hat{p}_{\{i,j\}}\}$.

In the SM process, we assume that in each round agents choose a matching with no *blocking pair* under these preferences (two agents who are not matched to each other but prefer each other to their matched partners). We additionally assume that once a pair of agents is successfully matched they will continue matching with each other in future rounds. It is straightforward to see that the *decentralized stable matching*, where there are no blocking pairs, can be formalized as follows.

Stable Matching (SM)

Initialize the set of successful edges $A \leftarrow \emptyset$. Let $S \leftarrow V$.

For rounds $1 \leq t \leq T$:

- While there are agents in S who are compatible with positive probability, find the most compatible pair f , match i with j , and remove i, j from S . Let M_t be the matching determined once no agents in S are compatible with positive probability.
- Output $M_t \cup A$ as the selected matching for round t .
- If an edge e in M_t is successful add it to A . Otherwise add both its endpoints back to S and set $p_e \leftarrow 0$.

In other words, the stable matching can be determined by greedily selecting pairs of agents who are most likely to be compatible, matching them, and committing to matching them in future rounds if their edge is successful. The stable matching is incentive-compatible *within-rounds*, in the sense that if in round t some agent i prefers matching with another agent j to their match in M_t then j is matched to a preferred agent. The stable matching is also incentive-compatible *across-rounds*, in the sense that agents who are successfully matched in previous rounds prefer staying matched to matching again, and agents always prefer to be matched as soon as possible.

For additional clarity, consider when all compatibility probabilities are distinct and strictly smaller than 1. In this case, the stable matching in the first round is unique: the pair with the highest compatibility match with each other (otherwise they would form a blocking pair), the pair with the highest probability among the remaining agents match, and so on. No matter the realizations, in all rounds there is a unique stable matching based on the updated preferences. When some edge probabilities are the same the stable matching may not be unique, but our analysis applies no matter which stable matching is selected, as long as previously matched edges stay matched.

We compare the result of SM with a platform that optimizes its matching process centrally. One downside to this approach is that determining this *optimal online algorithm (OPT)* is NP-hard.

PROPOSITION 1. *Computing the optimal online algorithm is NP-hard.*

This was originally proved by Chen et al. [9];¹ for completeness we provide a full proof for our setting in Appendix D. This hardness result motivates the analysis of matching algorithms which can be computed efficiently. Our main result is the following.

THEOREM 1. *The expected reward of stable matching is at least a 0.316-approximation of the expected reward of the optimal online algorithm.*

The proof of Theorem 1 relies on coupling edges selected by OPT with those selected by SM. A subset of these edges are bounded using a *Domination Lemma*, which uses the greediness of

¹[9] state the hardness result for a slightly different setting, where the optimal online algorithm is forced to commit to edges that are successful. Despite the fact that in our setting, OPT is not required to commit, the same proof holds.

SM to bound the expected reward generated by a subset of edges selected by OPT by twice the expected reward generated by SM. This coupling requires a delicate comparison of two algorithms that may have, through different histories, discovered very different pieces of information about the sample graph G . The remaining edges are bounded using a *Charging Lemma* that charges edges selected by OPT to adjacent edges selected by SM. The bounds provided by the Charging Lemma and the Domination Lemma define a factor-revealing LP that yields the 0.316-approximation. In Section 2.4, we additionally show an upper bound of $0.5 + \epsilon$ on the approximation ratio that SM achieves to OPT.

2.1 Coupling Edges Selected by Stable Matching and OPT

The main technical innovation in our paper is to couple the edges selected by OPT with the edges selected by SM in a way that admits a constant-factor bound in expectation for each round *despite the fact that SM and OPT learn different information in each round*. To specify the subsets of edges that are coupled in the Charging and Domination Lemmas, we introduce some additional notation.

Throughout the paper, for a random set S we will write $\mathbb{E}[S]$ as a shorthand for $\mathbb{E}[|S|]$. We use \sqcup to denote a union of sets that are disjoint.

For all $t \in [T]$, let $S_{\leq t}$ denote the set of *successful* edges selected by SM in round t .² Also define S_t to be the successful *new* edges that SM selects for the first time in round t . We hence have

$$S_{\leq t} = S_1 \sqcup S_2 \sqcup \dots \sqcup S_t. \quad (1)$$

Similarly, let $O_{\leq t}$ denote the set of *successful* edges that OPT selects in round t .

Fix t . To bound the expected number of edges in $O_{\leq t}$, we will partition these edges based on whether or not they can be added to $S_{\leq t}$. In particular, we write

$$O_{\leq t} = \text{AUG} \sqcup \text{ADJ} \quad (2)$$

where AUG denotes all the edges in $O_{\leq t}$ that are vertex-disjoint from $S_{\leq t}$ (and hence can augment this matching), and ADJ denotes the remaining edges that share an endpoint with some edge in $S_{\leq t}$.³ It will also be useful to categorize all edges in $O_{\leq t}$ according to the round in which OPT first selected them; hence, we will write

$$\text{AUG} = \text{AUG}_1 \sqcup \text{AUG}_2 \sqcup \dots \sqcup \text{AUG}_t, \quad (3)$$

where AUG_j denotes all edges in AUG that OPT first selected in round j . Similarly, we break up

$$\text{ADJ} = \text{ADJ}_1 \sqcup \text{ADJ}_2 \sqcup \dots \sqcup \text{ADJ}_t \quad (4)$$

where ADJ_j is all edges in ADJ that OPT first selected in round j . Note that AUG and ADJ and their corresponding partitions are defined with respect to $S_{\leq t}$ and $O_{\leq t}$ and therefore they all depend on t . We have dropped the index t from their notation to simplify the exposition.

In the rest of this section, we will use the Domination Lemma to couple edges in S_i with edges in AUG_i , and the Charging Lemma to couple edges in S_t with edges in ADJ_t . This coupling provides bounds that define the factor-revealing LP.

2.2 The Domination Lemma

Our main technical lemma is the Domination Lemma, which makes use of the greediness of SM to bound its approximation to OPT. In particular, recall that $S_{\leq i-1}$ is the set of *successful* edges selected by SM in round i . Then in each round $i \leq t$, SM selects edges to add on to $S_{\leq i-1}$ greedily.

²Note $S_{\leq t}$ is a random set that is a deterministic function of the sample graph G .

³As $S_{\leq t}$ and $O_{\leq t}$ are deterministic functions of the sample graph G , AUG and ADJ are hence also random sets entirely determined by the sample graph.

In this step, the Domination Lemma lower bounds its expected reward versus some of the edges selected by OPT.

LEMMA 1 (DOMINATION LEMMA). *Fix a round t and define AUG_i for $1 \leq i \leq t$ as in Equations (2) and (3). Then, for all $i \leq t$, $2 \cdot \mathbb{E}[S_i] \geq \mathbb{E}[\text{AUG}_i]$.*

PROOF. The expectations in the lemma are over all sample graphs. In the proof, we require a consideration of what each algorithm knows up to round i , and through this split up the probability space the expectations are being taken over. To do so, we define the notion of a history induced on an algorithm, and analyze the expected size of S_i and AUG_i conditioned on these histories.

Fix some $i \leq t$. For a sample graph G and an algorithm \mathcal{A} , we define $H^{\mathcal{A}}(G)$, the *history* that G induces on \mathcal{A} , as follows. The history $H^{\mathcal{A}}(G)$ consists of the set of matchings \mathcal{A} selects in the first $i - 1$ rounds, along with the observed outcomes $\{X_e\}$ of all the edges in these matchings, conditioned on the sample graph being G . We next partition the probability space over sample graphs based on the distinct histories they induce on SM in the first $i - 1$ rounds. In particular, for any possible history h , let \mathcal{G}_h denote the set of all sample graphs G such that $H^{\text{SM}}(G) = h$. We use $\mathbb{E}[S_i | \mathcal{G}_h]$ as a shorthand for the expected size of S_i , conditioned on the sample graph G being in \mathcal{G}_h , and use $\mathbb{P}[\mathcal{G}_h]$ for the probability that our sample graph G is an element of \mathcal{G}_h . Note then that

$$\mathbb{E}[S_i] = \sum_h \mathbb{E}[S_i | \mathcal{G}_h] \cdot \mathbb{P}[\mathcal{G}_h], \text{ and } \mathbb{E}[\text{AUG}_i] = \sum_h \mathbb{E}[\text{AUG}_i | \mathcal{G}_h] \cdot \mathbb{P}[\mathcal{G}_h],$$

where the sum is over all possible histories h . It hence suffices to show that $2 \cdot \mathbb{E}[S_i | \mathcal{G}_h] \geq \mathbb{E}[\text{AUG}_i | \mathcal{G}_h]$ for any history h .

To do so, we analyze $\mathbb{E}[\text{AUG}_i | \mathcal{G}_h]$ by partitioning \mathcal{G}_h based on the different histories these sample graphs induce on OPT. Specifically, for a fixed history h' , let $\mathcal{G}_{h,h'}$ denote all sample graphs G such that $H^{\text{SM}}(G) = h$ and $H^{\text{OPT}}(G) = h'$. Note that $\mathbb{E}[\text{AUG}_i | \mathcal{G}_h] = \sum_{h'} \mathbb{E}[\text{AUG}_i | \mathcal{G}_{h,h'}] \cdot \mathbb{P}[\mathcal{G}_{h,h'} | \mathcal{G}_h]$, and therefore, to complete the proof of Lemma 1, it suffices to show that

$$2 \cdot \mathbb{E}[S_i | \mathcal{G}_h] \geq \mathbb{E}[\text{AUG}_i | \mathcal{G}_{h,h'}]$$

for any possible histories h, h' . The rest of the proof is devoted to this inequality.

Conditioned on our sample graph being in $\mathcal{G}_{h,h'}$, let N be the *new* edges selected by OPT in round i that can augment the matching $S_{\leq i-1}$. As we mentioned earlier, we can assume that OPT is deterministic. So N is uniquely determined by h and h' . Also, by definition N is disjoint from all edges in h' and all successful edges in h .

Let $N_0 \subseteq N$ denote those edges in N that belong to neither h nor h' . Because edges are independent, for each edge e outside history h or h' , the probability that a random sample graph in $\mathcal{G}_{h,h'}$ includes edge e is exactly p_e . Therefore, the expected number of edges in N that are successful is given by $\sum_{e \in N_0} p_e$ as all edges in $N \setminus N_0$ are guaranteed to be unsuccessful.

Furthermore, observe that the set of edges in N_0 can feasibly augment $S_{\leq i-1}$ as every edge in N_0 is vertex-disjoint from $S_{\leq i-1}$, and the edges in N_0 form a matching disjoint from h . Note that SM chooses edges greedily to augment $S_{\leq i-1}$; as the greedy algorithm gives a $1/2$ -approximation to maximum weight matching, we hence have

$$2 \cdot \mathbb{E}[S_i | \mathcal{G}_h] \geq \sum_{e \in N_0} p_e.$$

Additionally, note that conditioned on our sample graph being in $\mathcal{G}_{h,h'}$, AUG_i is guaranteed to be a subset of the successful edges in N_0 , as edges augmenting $S_{\leq t}$ must also augment $S_{\leq i-1}$. Therefore,

$$\sum_{e \in N_0} p_e \geq \mathbb{E}[\text{AUG}_i | \mathcal{G}_{h,h'}].$$

We conclude

$$2 \cdot \mathbb{E}[S_i \mid \mathcal{G}_h] \geq \mathbb{E}[\text{AUG}_i \mid \mathcal{G}_{h,h'}]$$

as desired. \square

To obtain a bound of 0.316 on the approximation factor achieved by SM we provide a refinement of the Domination Lemma. To prove the Domination Lemma, roughly we showed that for any fixed round i , if we look at the successful edges that OPT selects for the first time in round i which can augment $S_{\leq i-1}$, the expected size of this set is no more than $2 \cdot \mathbb{E}[S_i]$. As AUG_i fits this description, this shows that $\mathbb{E}[\text{AUG}_i] \leq 2 \cdot \mathbb{E}[S_i]$. However, this lemma is loose; AUG_i does not necessarily comprise *all* the edges that OPT selects for the first time in round i which can augment $S_{\leq i-1}$. In particular, certain edges in ADJ_i might fit this description as well — they certainly cannot augment $S_{\leq i}$ by definition, but for small values of i many of them might be able to augment $S_{\leq i-1}$.

To refine the Domination Lemma, we categorize the edges in ADJ according to which of S_1, S_2, \dots, S_t they are incident to; this is motivated by the goal of finding some edges in ADJ_i that can augment S_i . In particular, we break up

$$\text{ADJ} = \text{ADJ}(S_1) \sqcup \text{ADJ}(S_2) \sqcup \dots \sqcup \text{ADJ}(S_t)$$

where $\text{ADJ}(S_j)$ denotes the subset of ADJ consisting of edges incident with S_j but not $S_{\leq j-1}$. This is well-defined because the sets S_1, S_2, \dots, S_t are disjoint and form a matching. Similarly, for any fixed j we break up

$$\text{ADJ}(S_j) = \text{ADJ}_1(S_j) \sqcup \text{ADJ}_2(S_j) \sqcup \dots \sqcup \text{ADJ}_t(S_j)$$

where $\text{ADJ}_i(S_j)$ is all edges in $\text{ADJ}(S_j)$ that OPT first selected in round i .

The informal statement of the refined Domination Lemma is that if we fix a round i , and look at all edges that OPT first discovered in round i that can be added to $S_{\leq j-1}$, the expected size of this set is at most twice the expected size of S_j . Our notation above lets us write this formally.

LEMMA 2 (REFINED DOMINATION LEMMA). *Fix a round t . We have for all $i, j \leq t$ that*

$$\mathbb{E}[\text{AUG}_i] + \mathbb{E}[\text{ADJ}_i(S_j)] + \mathbb{E}[\text{ADJ}_i(S_{j+1})] + \dots + \mathbb{E}[\text{ADJ}_i(S_t)] \leq 2 \cdot \mathbb{E}[S_j].$$

The proof of the Refined Domination Lemma only requires small changes from the proof of the Domination Lemma, and can be found in Appendix E.1.

2.3 Proof of Theorem 1

We next state and prove the Charging Lemma, related to the fact that two maximal matchings in a graph always have sizes within a factor of 2.

LEMMA 3 (CHARGING LEMMA). *Fix a round t . For all $j \leq t$,*

$$\sum_{i=1}^t \mathbb{E}[\text{ADJ}_i(S_j)] \leq 2 \cdot \mathbb{E}[S_j].$$

PROOF. Fix a sample graph. Consider any edge $e \in \cup_i \text{ADJ}_i(S_j)$. By definition, it is incident to at least one edge in S_j ; charge e to one of the edges in S_j it is incident to. Note that because $\cup_i \text{ADJ}_i(S_j)$ forms a matching, every edge in S_j is charged at most twice. Hence the result holds sample graph by sample graph, as well as in expectation over all sample graphs. \square

The task remaining is to use the structure we have found to give an upper bound on $\frac{\mathbb{E}[O_{\leq t}]}{\mathbb{E}[S_{\leq t}]}$; we do so via a factor-revealing linear program, with constraints corresponding to the Refined Domination Lemma and the Charging Lemma.

Factor-revealing LP (Primal)

$$\max \quad \sum_{i=1}^t X_i + \sum_{i=1}^t \sum_{j=1}^t X_{i,j}$$

$$\text{such that} \quad X_i + \sum_{q=j}^t X_{i,q} \leq 2Y_j \quad \text{for all } 1 \leq i, j \leq t \quad (1)$$

$$\sum_{i=1}^t X_{i,j} \leq 2Y_j \quad \text{for all } 1 \leq j \leq t \quad (2)$$

$$\sum_{j=1}^t Y_j \leq 1 \quad (3)$$

$$Y_j \geq 0, \quad X_i \geq 0, \quad X_{i,j} \geq 0 \quad \text{for all } 1 \leq i, j \leq t \quad (4)$$

In this linear program, the variables $\{X_i\}_{1 \leq i \leq t}$ correspond to $\{\mathbb{E}[\text{AUG}_i]\}$, the variables $\{X_{i,j}\}_{1 \leq i, j \leq t}$ correspond to $\{\mathbb{E}[\text{ADJ}_i(S_j)]\}$, and the variables $\{Y_j\}_{1 \leq j \leq t}$ correspond to $\{\mathbb{E}[S_j]\}$, and the bounds from the lemmas are encoded as constraints.

Given any instance of our matching problem, we can construct a feasible solution for this LP by setting:

$$X_i = \frac{\mathbb{E}[\text{AUG}_i]}{\mathbb{E}[S_{\leq t}]}, \quad X_{i,j} = \frac{\mathbb{E}[\text{ADJ}_i(S_j)]}{\mathbb{E}[S_{\leq t}]}, \quad Y_j = \frac{\mathbb{E}[S_j]}{\mathbb{E}[S_{\leq t}]}.$$

Indeed, note that when we set the variables in this way, (1) directly states the Refined Domination Lemma, and (2) directly states the Charging Lemma. Also, because $\sum_j |S_j| = |S_{\leq t}|$ we have that (3) holds. For the suggested feasible solution, we can note that the objective simplifies to:

$$\sum_{i=1}^t X_i + \sum_{i=1}^t \sum_{j=1}^t X_{i,j} = \frac{\sum_i \mathbb{E}[\text{AUG}_i] + \sum_{i,j} \mathbb{E}[\text{ADJ}_i(S_j)]}{\mathbb{E}[S_{\leq t}]} = \frac{\mathbb{E}[O_{\leq t}]}{\mathbb{E}[S_{\leq t}]}$$

Hence, the maximum objective obtained by our LP gives a lower bound on the worst-case competitive ratio that SM achieves against OPT.

To give a bound on the maximum value obtained by this LP, we take the dual, noting that by weak duality it suffices to analyze the value obtained by a specific feasible solution. The dual of our LP is given below.

Factor-revealing LP (Dual)

$$\min \quad u$$

$$\text{such that} \quad \sum_{q=1}^j F_{i,q} + c_j \geq 1 \quad \text{for all } 1 \leq i, j \leq t \quad (1)$$

$$\sum_{i=1}^t 2F_{i,j} + 2c_j \leq u \quad \text{for all } 1 \leq j \leq t \quad (2)$$

$$\sum_{j=1}^t F_{i,j} \geq 1 \quad \text{for all } 1 \leq i \leq t \quad (3)$$

$$u \geq 0, \quad c_j \geq 0, \quad F_{i,j} \geq 0 \quad \text{for all } 1 \leq i, j \leq t \quad (4)$$

Consider a feasible solution for the dual LP given by $F_{i,j} = \frac{1}{1+t(e-1)} \cdot \left(\frac{t}{t-1}\right)^{j-1}$ for $1 \leq i, j \leq t$, $c_j = 1 - \frac{t^j - (t-1)^j}{t(t-1)^{j-1}e - (t-1)^j}$ for $1 \leq j \leq t$ and $u = \max_{j \in [t]} \left\{ \sum_i 2F_{i,j} + 2c_j \right\} = 2 + \frac{2(t-1)^t}{t^t - (t-1)^t}$.⁴ The objective value $u = u(t)$ is increasing in t and

$$\lim_{t \rightarrow \infty} u(t) = 2 + \frac{2}{e-1}.$$

Hence by weak duality, the objective value of our primal is at most $2 + \frac{2}{e-1}$. It follows that the expected reward of SM in round t is at least a $\left(2 + \frac{2}{e-1}\right)^{-1} \geq 0.316$ fraction of the expected reward of OPT in round t . This completes the proof of Theorem 1.

2.4 Discussion: How much better can a matching platform do?

So far, we have shown that a decentralized stable matching process achieves at least a 0.316 proportion of the reward of the optimal matching service. In this section, we provide some additional discussion on what else is achievable by the platform.

First, suppose the platform could invest in an off-the-shelf matching process. In particular, we consider the GREEDY-COMMIT algorithm that was previously proposed by [9]. The GREEDY-COMMIT algorithm proceeds by proposing in the first round a matching of vertices in V that maximizes the sum of edge probabilities, and in subsequent rounds committing to keeping all successful edges from the previous rounds, and augmenting them with a matching that maximizes the sum of edge probabilities in the remaining graph. We describe GREEDY-COMMIT formally in Appendix B.

Note that GREEDY-COMMIT is computable in polynomial-time, and is incentive-compatible across rounds but not within rounds. Hence GREEDY-COMMIT can be thought of as a matching service that *hides information* and myopically dictates a maximum matching in each round, but cannot incentivize matched agents to be rematched in future rounds.

Chen et al. showed that GREEDY-COMMIT gives a 0.25-approximation to OPT.⁵ As an extension to our other results in this section, we show that our techniques improve on the analysis of Chen et al. In particular, we show the following result.

THEOREM 2. *The expected reward of GREEDY-COMMIT is at least a 0.43-approximation to OPT.*

The proof proceeds very similarly to the proof of the approximation factor for SM. The Charging Lemma continues to hold, and a stronger version of the Refined Domination Lemma holds, improved by a factor of 2. These constraints yield a different factor-revealing LP which we use to prove the claimed 0.43 factor. Details can be found in Appendix E.2.

Another natural question is what upper bound we can give on how well SM and GREEDY-COMMIT can perform compared to the optimum online algorithm. Although computing OPT is NP-hard, perhaps one of them could be very close to optimal, in the sense of guaranteeing a $(1-\varepsilon)$ -approximation to OPT for some small ε ? We show this is not the case; in particular, we show that both SM and OPT provide at best a $1/2$ -approximation to the optimal online algorithm.

LEMMA 4. *The expected reward of SM and GREEDY-COMMIT are at most a $1/2$ -approximation to OPT.*

⁴This is the optimal Dual solution. Note that if we change our objective, we might be able to get better approximation factors; e.g. if we change the objective to be the cumulative reward, which sums the successful selected edges over every round instead of finding a round-by-round approximation guarantee, we will be able to achieve a 2.96-approximation factor.

⁵[9] provide a proof for a slightly different result which works to prove Lemma 4; see footnote 1.

PROOF. Consider the bipartite graph $G_{n,\epsilon}$ with bipartition $(U = (u_1, \dots, u_n), V = (v_1, \dots, v_n))$ edges $E = \{u_1v_i\} \cup \{u_i v_1\}$, the edge u_1v_1 has probability 1 of being successful, and every other edge has probability $0.5 - \epsilon$. We illustrate the graph in Figure ?? in the appendix.

Suppose there are $T = n^2$ rounds. SM matches u_1 to v_1 in the first round. Since the edge is successful, u_1 will be matched to v_1 in every subsequent round. Hence SM obtains a reward of n^2 .

We now define a different algorithm that selects the matching $(u_1v_{i+1}) \cup (u_{i+1}v_1)$ in round i , for $1 \leq i \leq n - 1$. With probability $1 - o(1)$, both edges are successful in at least one of these rounds. In this case the algorithm can select this matching from round n to round T , obtaining payoff $2n^2(1 - o(1))$ in these rounds. In expectation, the reward of the above algorithm is at least $(1 - o(1)) \cdot 2n^2(1 - o(1)) = 2n^2(1 - o(1))$. As SM only obtains a reward of n^2 , taking n sufficiently large proves the claim.

The same example proves that GREEDY-COMMIT is at most a $(1/2 + \epsilon)$ -approximation to OPT. \square

Finally, one might wonder how much of this gap between stable matching or GREEDY-COMMIT and OPT is due to the fact that both algorithms commit to previously matched successful pairs, while OPT does not. Consider the optimal online algorithm that is restricted to committing to successfully matched pairs (*OPT-Commit*). OPT-Commit can be thought of as the optimal matching service that improves on GREEDY-COMMIT by dictating a matching in each round in an optimal *non-myopic* manner, but again cannot incentivize matched agents to be rematched in future rounds. We note that the expected reward achieved by OPT-Commit is not always the same as OPT (see Appendix C), so committing causes some loss of reward.

Furthermore, we show that OPT-Commit achieves at least a 0.5-approximation to OPT. Since SM and GREEDY-COMMIT achieve at best a 0.5-approximation to OPT, if they do not achieve this then part of the loss is due to the fact that both SM and GREEDY-COMMIT are *myopic* and do not account for the effect of current queries on possible matchings in future rounds, rather than the fact that both commit to previously matched successful pairs.

PROPOSITION 2. *The expected reward of OPT-commit is at least a $1/2$ -approximation to OPT.*

The proof of Proposition 2 relies on constructing an algorithm that commits to successfully matched pairs, and that is a $1/2$ -approximation to OPT. The algorithm (which we denote by \mathcal{A}) proceeds as follows: in round i , \mathcal{A} selects all the successful edges it has found so far, as well as the edges that OPT selects in round i which can augment this matching. We show that in every round the expected size of the matching found by \mathcal{A} is at least a $1/2$ -approximation to the expected size of the matching selected by OPT. We provide a complete proof of Proposition 2 in Appendix C.

In sum, we have considered a number of different matching algorithms. SM is decentralized, while the rest are *centralized* matching platforms that determine and prescribe either a myopic maximum matching (GREEDY-COMMIT) or a non-myopic optimal matching (OPT-Commit, OPT). All of the centralized matching platforms must hide some amount of information from participants, as the prescribed matchings in each round are not necessarily stable and hence not within-round incentive-compatible. In the case of OPT the matching proposed by the platform in addition is not across-round compatible, as it needs to coerce successfully matched agents to return and be rematched in future rounds. Both types of incentive constraints are not without loss, and in addition in order to implement either OPT-Commit and OPT the platform must solve an NP-hard problem. Nonetheless all of these platform designs attain an expected reward within a constant-factor approximation of OPT.

3 CAPACITATED MATCHING

In this section, we consider the difference between stable matching and optimal online matching in more general settings, such as where all agents can be in bilateral matches with multiple of any of the other agents, or agents can be grouped in teams. For example, styling and clothing rental companies may have limited inventory of different items to send to customers, and students may be able to take on multiple projects, or may be grouped into project teams.

Formally, we have a general graph $G(V, E)$ where vertices have capacities $\{C(v)\}_{v \in V}$ given by arbitrary positive integers. In each of T rounds, we can select a capacitated matching among these agents, i.e. some $E' \subseteq E$ such that each v has at most $C(v)$ edges in E' incident to it. As in the previous section, each agent's v 's goal is to maximize the expected number of successful matches they participate in over all rounds, and the platform's goal is to maximize the weighted sum of the rewards collected in all T rounds.

How can we define an appropriate decentralized outcome in this general setting? Suppose agents can observe their compatibility $p_{\{i,j\}}$ with every other agent. A similar argument to the previous section shows that each agent i maximizes their individual expected number of successful matches by matching in each round with the $C(v)$ agents who are most likely to be compatible with i (including agents to whom i has been successfully matched in the past). Hence, in the decentralized matching, there again will be no *blocking pair* of two agents who are not matched to each other but both prefer the other to at least one of their matched partners. Moreover, once a pair of agents is successfully matched they will continue matching with each other in future rounds. This determines a decentralized capacitated stable matching process, which we will again denote by SM.

In this general setting, we show that SM gives a constant-factor approximation to OPT.

THEOREM 4. *In the setting where agents have arbitrary capacities, the expected reward achieved by the SM algorithm is at least a $1/11$ fraction of the expected reward of the optimal online algorithm.*

Moreover, in many practical matching settings (e.g. gig economy applications, mentorship) matches occur between two sides of a market, and agents on one side of the market (e.g. jobs, mentees) only have capacity 1. In this setting, we can make use of the structure of the underlying graph to refine our approximation factor.

THEOREM 5. *In the many-to-one setting, the expected reward obtained by SM is at least a $1/7$ fraction of the expected reward of the optimal online algorithm.*

Next we consider a setting where for some set of agents V , each subset $S \subseteq V$ of size at most k has an associated probability p_S of being “compatible” as a team. Generalizing the previous setting, in each of T rounds, the agents are partitioned into teams so that each agent is in at most one team. We can equivalently think of the problem as constructing round-by-round matchings in a random hypergraph. We use modifications of the Domination and Charging Lemmas to show that the decentralized greedy formation of teams of constant size provides a constant-factor approximation to the optimal online algorithm. Formally, we show the following result, and provide a full proof in Appendix E.5.

THEOREM 6. *In a hypergraph where all hyperedges have cardinality at most k , the expected reward obtained by SM is at least $1/2k$ of the expected reward of OPT.*

3.1 What changes with capacities?

We first provide some intuition for what differentiates the capacitated setting from the setting in Section 2, where agents can only be matched with one other agent every round. This intuition will

motivate the new proof techniques used in the capacitated setting, and also highlight the nuances in the decomposition used to prove Theorem 1.

The difficulty in extending the techniques in Section 2 lies in the definition of what it means to be an augmenting edge, which in turn determines the decomposition of the edges OPT selects into AUG and ADJ. A crucial property underlying the proof of the domination lemma is that in the non-capacitated setting, whether a matching N augments another matching N' can be determined by checking whether each $e \in N$ augments N' .⁶ Specifically, if each $e \in N$ individually can augment N' , and the edges in N' form a matching, then *all* edges in N' can jointly augment N . For natural definitions of augmentation in the capacitated setting (e.g. N augments N' if $N \sqcup N'$ is a capacitated matching), a statement of this sort is no longer true.

Hence, it is natural that in the capacitated case we might give a definition of AUG_i constructed by jointly considering a group of edges in AUG_i that can augment $S_{\leq i-1}$, instead of considering whether edges in AUG_i could augment $S_{\leq i-1}$ individually. In particular, we might define AUG_i to be a maximum subset of edges in O_i which can augment $S_{\leq i-1}$, and hope that we could claim $\mathbb{E}[\text{AUG}_i] \leq C \cdot \mathbb{E}[S_{\leq i-1}]$ for some constant C . This natural approach unfortunately fails; the reason is that AUG_i can only be formed using information that SM did not have available at the start of round i .⁷

Given that these natural generalizations to the augmentation approach fail, we instead decompose successful edges OPT selects based on a notion of occupancy for each agent. We say that a node is *heavily occupied* if at least half their capacity is taken up by edges in $S_{\leq t}$. We decompose $O_{\leq t}$, the successful edges that OPT selects in round t , into edges that are in $S_{\leq t}$, edges that are not in $S_{\leq t}$ but have at least one of their endpoints heavily occupied by $S_{\leq t}$, and the remaining edges. Formally, we write

$$O_{\leq t} = (O_{\leq t} \cap S_{\leq t}) \sqcup \text{Occ} \sqcup \text{REM}$$

where Occ denotes all edges in $O_{\leq t} \setminus S_{\leq t}$ with an endpoint that is filled to at least half-capacity from edges in $S_{\leq t}$, and REM denotes the remaining edges in $O_{\leq t} \setminus (S_{\leq t} \sqcup \text{Occ})$.

Later in Subsection 3.3, we show a tighter analysis of this same decomposition technique for bipartite many-to-one matchings results in a better approximation factor. In both Sections 3.2 and 3.3, we use a generalized version of the Charging Lemma to bound edges in Occ incident with nodes that are heavily occupied, and a generalized version of the Domination Lemma to bound edges in REM incident only with nodes that are not heavily occupied.

3.2 A $1/11$ -approximation for Capacitated Matchings on General Graphs

In this section, we prove Theorem 4, and show that SM achieves at least a $1/11$ fraction of the expected reward of OPT in the general capacitated setting.

Recall that $A_{\leq t}$ and $O_{\leq t}$ denote the *successful* edges selected by SM and OPT, respectively, in round t . The proof of Theorem 4 will follow from proving for all $t \in [T]$

$$\mathbb{E}[O_{\leq t}] \leq 11 \cdot \mathbb{E}[S_{\leq t}]. \quad (5)$$

Recall that we decomposed $O_{\leq t} = (O_{\leq t} \cap S_{\leq t}) \sqcup \text{Occ} \sqcup \text{REM}$. This decomposition immediately lets us write a generalized Charging Lemma.

LEMMA 5 (GENERALIZED CHARGING LEMMA). *We have $\mathbb{E}[\text{Occ}] \leq 4 \cdot \mathbb{E}[S_{\leq t}]$.*

⁶An edge e augments N' if and only if it doesn't share any endpoints with edges in N' , a matching N augments N' if and only if it doesn't share any endpoints with edges in N' .

⁷In the previous domination lemma, the fact that we could augment edge-by-edge let us argue that AUG_i could actually be bounded only using information that SM had available at round i . This subtlety reinforces the need for the careful conditioning in the previous proof.

PROOF. Fix any sample graph G . For each edge $e \in \text{Occ}$, charge it to an endpoint of e with at least half of its capacity occupied by edges in $S_{\leq t}$. As Occ forms a valid capacitated matching, the charge placed on each vertex is at most its capacity. Additionally, note that we can cover each of these charges by having every edge in $S_{\leq t}$ pay two charges to each of its endpoints. The result follows by averaging over all sample graphs. \square

The main claim for which we require different ideas is a generalized Domination Lemma. Recall that S_t and O_t denote the *new successful* edges selected in round t by SM and OPT, respectively. As before, we need to analyze the successful edges selected by OPT based on the round they were discovered; with this in mind we write

$$\text{REM} = \text{REM}_1 \sqcup \text{REM}_2 \sqcup \dots \sqcup \text{REM}_t$$

where REM_i denotes those edges in REM which OPT selected for the first time in round i . The Generalized Domination Lemma bounds REM_i against S_i .

LEMMA 6 (GENERALIZED DOMINATION LEMMA). *For all $i \leq t$ we have $\mathbb{E}[\text{REM}_i] \leq 6 \cdot \mathbb{E}[S_i]$.*

PROOF. Using the same notation in the previous proof of the Domination Lemma, for fixed histories h, h' it suffices to show that

$$\mathbb{E}[\text{REM}_i \mid \mathcal{G}_{h,h'}] \leq 6 \cdot \mathbb{E}[S_i \mid \mathcal{G}_h].$$

Conditioned on our sample graph being in $\mathcal{G}_{h,h'}$, there is a fixed set N of edges that OPT selects in round i which are not in $S_{\leq i-1}$, and furthermore have neither endpoint filled to half-capacity by edges in $S_{\leq i-1}$. Using the same logic as before, we have that if $N_0 \subseteq N$ denotes those edges in N disjoint from h and h' , then

$$\mathbb{E}[\text{REM}_i \mid \mathcal{G}_{h,h'}] \leq \sum_{e \in N_0} p_e$$

as REM_i is some (random) subset of N_0 . In contrast to previous settings, it is not the case that the set of edges in N_0 can feasibly augment $S_{\leq i-1}$. However, we show there exists a subset of edges from N_0 with at least one-third of the total weight which can feasibly augment $S_{\leq i-1}$. To prove this, we need the following graph theoretic claim.

CLAIM 1. *In any weighted graph $G = (V, E)$ where each vertex v has degree d_v , there exists a subgraph S whose edges contain at least $1/3$ of the total weight of all edges in G , such that each vertex v has degree at most $\lceil \frac{d_v}{2} \rceil$ in S .*

PROOF. Consider the following algorithm for constructing S . Of all remaining edges in E , add the one of maximum weight to S . Call this edge $e = (u, v)$; remove e from G . Furthermore, if u has remaining incident edges in G , remove one (chosen arbitrarily). If v has remaining incident edges in G , remove one (chosen arbitrarily). Continue until G has no edges remaining.

Note that for any vertex v , there are at most $\lceil \frac{d_v}{2} \rceil$ edges of S incident to v . Indeed, after adding any edge to S that is incident to v , we delete an edge in G incident to v if possible. Note also that in each step, the weight of the edge we add to S is at least the weight of each edge we delete. Hence in each step, we add to S at least a $1/3$ fraction of the weight of the edge we added and the edges we deleted. This holds in aggregate over all rounds. \square

To finish the proof of the lemma, we use this claim to find a subgraph S of (V, N_0) with at least $1/3$ of the weight; we claim this subgraph is feasible to augment $S_{\leq i-1}$ (and can be selected by SM in round i). Indeed, note it is disjoint from h , and each vertex v is filled to at most $\lceil \frac{C(v)}{2} \rceil$ capacity in S (because N_0 is a valid capacitated matching). Furthermore, every endpoint v of an edge in S is filled to at most $\lfloor \frac{C(v)}{2} \rfloor$ capacity by edges in $S_{\leq i-1}$. Thus, S is a feasible set of edges to augment

$S_{\leq i-1}$ in round i . Well-known results also imply that in a fixed round, the SM algorithm gives a $1/2$ -approximation to maximum weight augmenting edges (see, e.g., [39]). Hence,

$$\mathbb{E}[S_i \mid \mathcal{G}_h] \geq \sum_{e \in S} p_e \geq \frac{1}{6} \cdot \sum_{e \in N_0} p_e.$$

This demonstrates the result. \square

From Lemma 6 we can see

$$\mathbb{E}[\text{REM}] = \sum_{i=1}^t \mathbb{E}[\text{REM}_i] \leq 6 \sum_{i=1}^t \mathbb{E}[S_i] \leq 6 \cdot \mathbb{E}[S_{\leq t}].$$

Hence, using both Lemmas 5 and Lemma 6 we have

$$\mathbb{E}[O_{\leq t}] = \mathbb{E}[O_{\leq t} \cap S_{\leq t}] + \mathbb{E}[\text{Occ}] + \mathbb{E}[\text{REM}] \leq 11 \cdot \mathbb{E}[S_{\leq t}]$$

which completes the proof of Theorem 4.

3.3 A $1/7$ -approximation for Many-to-One Matchings

In many practical matching applications, matches occur between two sides of a market, where agents on one side can accept multiple matches while agents on the other side can be matched at most once. For example, workers may take on multiple jobs which each only need one worker, and mentors are frequently matched with multiple mentees who each only have one mentor. In this section, we consider the performance of SM in this more restricted setting of many-to-one bipartite matchings. In particular, we show our previous decomposition achieves a better approximation factor.

Formally, we assume our agents are broken into two disjoint sets $S \sqcup M$ (think “students” and “mentors”); for each pair (s, m) where $s \in S$ and $m \in M$ we are given a “success probability” $p_{s,m}$ representing the chance that an edge between s and m is realized when nature samples a random graph. Vertices in S are on the left and vertices in M are on the right. Each agent $m \in M$ has capacity $C(m) \in \mathbb{Z}_{>0}$, and each agent $s \in S$ has capacity $C(s) = 1$. We show Theorem 5, which we restate here.

THEOREM 5. *In the many-to-one setting, the expected reward obtained by SM is at least a $1/7$ fraction of the expected reward of the optimal online algorithm.*

PROOF. The proof will proceed by showing $\mathbb{E}[O_{\leq t}] \leq 7 \cdot \mathbb{E}[S_{\leq t}]$ for all $t \in [T]$. Partition the edges of $O_{\leq t}$ as

$$O_{\leq t} = \text{Occ} \sqcup \text{REM}$$

where Occ denotes edges in $O_{\leq t}$ whose left endpoint is incident to some edge in $S_{\leq t}$, or whose right endpoint is occupied to at least half capacity by edges in $S_{\leq t}$, and REM denotes the remaining edges in $O_{\leq t}$.⁸

LEMMA 7 (MANY-TO-ONE CHARGING LEMMA). *We have $\mathbb{E}[\text{Occ}] \leq 3 \cdot \mathbb{E}[S_{\leq t}]$.*

PROOF. Fix any sample graph. For each edge in Occ , if it is adjacent to an edge in $S_{\leq t}$ along its left endpoint, charge it to the unique edge in $S_{\leq t}$ it is adjacent to. Otherwise, place a charge on its right endpoint. Each right vertex v that is charged by this process is charged at most $C(v)$ times by this process, and has at least $C(v)/2$ edges in $S_{\leq t}$ incident to it. So the charges on right vertices can be covered by charging each edge in $S_{\leq t}$ at most twice. In all, we have charged each edge in $S_{\leq t}$ at most three times; the result follows by averaging over all sample graphs. \square

⁸Note this is an extremely similar decomposition to that in the previous section. We do not have a separate category for edges in $O_{\leq t}$ that are also in $S_{\leq t}$, as every $e \in O_{\leq t} \cap S_{\leq t}$ is automatically in Occ .

LEMMA 8 (MANY-TO-ONE DOMINATION LEMMA). *For all $i \leq t$ we have $\mathbb{E}[\text{REM}_i] \leq 4 \cdot \mathbb{E}[S_i]$.*

PROOF. The intuition behind this lemma is that at round i , there exists a feasible subset of the edges in REM_i that augments $S_{\leq i-1}$, with at least half the total weight. As before, the greedy property of SM algorithm gives a $1/2$ -approximation to maximum weight matching on a round-by-round basis. The rigorous proof follows the structure of previous domination lemmas very closely; details are available in Appendix E.5. \square

Combining Lemma 7 and Lemma 8 gives

$$\mathbb{E}[O_{\leq t}] = \mathbb{E}[\text{Occ}] + \sum_{i=1}^t \mathbb{E}[\text{REM}_i] \leq 3 \cdot \mathbb{E}[S_{\leq t}] + \sum_{i=1}^t 4 \cdot \mathbb{E}[S_i] = 7 \cdot \mathbb{E}[S_{\leq t}],$$

which completes the proof. \square

4 CONCLUSION AND FUTURE DIRECTIONS

This paper contributes to the literature on centralized and decentralized matching in platforms by providing an additional justification for focusing on decentralized algorithms. In particular, we consider matching platforms with repeated interactions between long-lived agents who have unknown but persistent preferences, such as gig economy applications, mentorship matching, kidney exchange, and team formation. We show that letting demand and supply myopically reach a stable matching in a decentralized manner approximates the outcome of computing and imposing a centralized matching. This is despite the fact that the centralized matching processes we consider are not incentive-compatible, and may even be NP-hard to compute.

We focused on a setting for matching with stochastic rewards and learning dynamics. While this setting has been studied in the query-commit literature, the primary motivation in that literature was failure-aware kidney exchange, and the primary focus of that literature was on querying individual edges in sequence to optimize some last-round objective. Beyond kidney exchange, this setting reflects natural learning dynamics that are present in a wide variety of markets that repeatedly match the same agents and provide persistent rewards, such as gig economy job markets, mentorship programs and team formation, and our focus on querying *matchings* and maximizing rewards received *in all rounds* is motivated by these settings. Further work can be done to identify when techniques from one approach can be transferred to the other.

Another difference from the query-commit literature is that we ask how a decentralized greedy algorithm compares to an optimal online algorithm that is *not restricted to commit* to past successes. While we show that OPT-Commit is a 2-approximation to OPT, further characterization of the relationship between OPT and OPT-Commit remains open. In addition, our results hold for any objective that takes a weighted sum of rewards across rounds. This is because the analysis throughout the paper was performed on a round-by-round basis. Objectives of potential interest included the sum of rewards across different rounds, expected discounted rewards, as well as the size of maximum successful matching that can be identified at the end of T rounds, and we believe our analysis can be tightened for these specific objectives to produce sharper bounds. We also believe that there are sharper bounds for the capacitated settings. We leave the study of tighter approximation guarantees to future work.

More broadly, our findings raise a number of follow-up questions. Can platforms be designed in a way to nudge agents towards a decentralized outcome that approximates the optimal centralized matching achievable by the platform? What is the approximation gap between incentive-compatible matching platforms and what is achievable by a centralized authority? This paper also explores the problem of learning through prior assignments, a natural direction that is relatively

less studied in the literature. Variations on this theme merit future exploration; for example, matching programs in highly relational and idiosyncratic settings like mentorship matching frequently face a *cold start* problem, where the program designer has some prior over which features best predict a successful match, and can refine this prior by observing the outcomes of prior matches. In mentorship or rotation programs an additional feature is that edge successes are often *correlated*; for example a student may think she has a general interest in economics and computation but learn that she is more interested in theory generally. We hope that this paper will motivate future work in these directions.

REFERENCES

- [1] Marek Adamczyk. Improved analysis of the greedy algorithm for stochastic matching. *Information Processing Letters*, 111(15):731–737, 2011.
- [2] Marek Adamczyk, Maxim Sviridenko, and Justin Ward. Submodular stochastic probing on matroids. *Mathematics of Operations Research*, 41(3):1022–1038, 2016.
- [3] Mohammad Akbarpour, Shengwu Li, and Shayan Oveis Gharan. Thickness and information in dynamic matching markets. *Journal of Political Economy*, 128(3):783–815, 2020.
- [4] Itai Ashlagi, Maximilien Burq, Patrick Jaillet, and Vahideh Manshadi. On matching and thickness in heterogeneous dynamic markets. *Operations Research*, 67(4):927–949, 2019.
- [5] Siddhartha Banerjee, Daniel Freund, and Thodoris Lykouris. Pricing and optimization in shared vehicle systems: An approximation framework. *arXiv preprint arXiv:1608.06819*, 2016.
- [6] Siddhartha Banerjee, Ramesh Johari, and Carlos Riquelme. Pricing in ride-sharing platforms: A queueing-theoretic approach. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, pages 639–639, 2015.
- [7] Nikhil Bansal, Anupam Gupta, Jian Li, Julián Mestre, Viswanath Nagarajan, and Atri Rudra. When lp is the cure for your matching woes: Improved bounds for stochastic matchings. *Algorithmica*, 63(4):733–762, 2012.
- [8] Dimitri P Bertsekas. *Dynamic programming and optimal control*, volume 1. Athena scientific Belmont, MA, 1995.
- [9] Ning Chen, Nicole Immorlica, Anna R Karlin, Mohammad Mahdian, and Atri Rudra. Approximating matches made in heaven. In *International Colloquium on Automata, Languages, and Programming*, pages 266–278. Springer, 2009.
- [10] Nikhil R Devanur and Thomas P Hayes. The adwords problem: online keyword matching with budgeted bidders under random permutations. In *Proceedings of the 10th ACM conference on Electronic commerce*, pages 71–78, 2009.
- [11] John P Dickerson, Ariel D Procaccia, and Tuomas Sandholm. Failure-aware kidney exchange. In *Proceedings of the fourteenth ACM conference on Electronic commerce*, pages 323–340, 2013.
- [12] Jon Feldman, Aranyak Mehta, Vahab Mirrokni, and Shan Muthukrishnan. Online stochastic matching: Beating 1-1/e. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 117–126. IEEE, 2009.
- [13] Yi Gai, Bhaskar Krishnamachari, and Rahul Jain. Learning multiuser channel allocations in cognitive radio networks: A combinatorial multi-armed bandit formulation. In *2010 IEEE Symposium on New Frontiers in Dynamic Spectrum (DySPAN)*, pages 1–9. IEEE, 2010.
- [14] Buddhima Gamkall, Sagar Kale, and Ola Svensson. Beating greedy for stochastic bipartite matching. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2841–2854. SIAM, 2019.
- [15] John Gittins. A dynamic allocation index for the sequential design of experiments. *Progress in statistics*, pages 241–266, 1974.
- [16] Gagan Goel and Aranyak Mehta. Online budgeted matching in random input models with applications to adwords. In *SODA*, volume 8, pages 982–991, 2008.
- [17] Gagan Goel and Pushkar Tripathi. Matching with our eyes closed. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 718–727. IEEE, 2012.
- [18] Vineet Goyal and Rajan Udwani. Online matching with stochastic rewards: Optimal competitive ratio via path based formulation. *arXiv preprint arXiv:1905.12778*, 2019.
- [19] Anupam Gupta and Viswanath Nagarajan. A stochastic probing problem with applications. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 205–216. Springer, 2013.
- [20] Anupam Gupta, Viswanath Nagarajan, and Sahil Singla. Algorithms and adaptivity gaps for stochastic probing. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 1731–1747. SIAM, 2016.
- [21] Anupam Gupta, Viswanath Nagarajan, and Sahil Singla. Adaptivity gaps for stochastic probing: Submodular and xos functions. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1688–1702. SIAM, 2017.
- [22] Itai Gurvich and Amy Ward. On the dynamic control of matching queues. *Stochastic Systems*, 4(2):479–523, 2015.
- [23] Ming Hu and Yun Zhou. Dynamic type matching. *Rotman School of Management Working Paper*, (2592622), 2020.

- [24] Zhiyi Huang and Qiankun Zhang. Online primal dual meets online matching with stochastic rewards: configuration lp to the rescue. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1153–1164, 2020.
- [25] Patrick Jaillet and Xin Lu. Online stochastic matching: New algorithms with better bounds. *Mathematics of Operations Research*, 39(3):624–646, 2014.
- [26] Ramesh Johari, Vijay Kamble, and Yash Kanoria. Matching while learning. *arXiv preprint arXiv:1603.04549*, 2016.
- [27] Chinmay Karande, Aranyak Mehta, and Pushkar Tripathi. Online bipartite matching with unknown distributions. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 587–596, 2011.
- [28] Richard M Karp, Umesh V Vazirani, and Vijay V Vazirani. An optimal algorithm for on-line bipartite matching. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 352–358, 1990.
- [29] Thomas Kesselheim, Klaus Radke, Andreas Tönnis, and Berthold Vöcking. An optimal online algorithm for weighted bipartite matching and extensions to combinatorial auctions. In *European symposium on algorithms*, pages 589–600. Springer, 2013.
- [30] Nitish Korula and Martin Pál. Algorithms for secretary problems on graphs and hypergraphs. In *International Colloquium on Automata, Languages, and Programming*, pages 508–520. Springer, 2009.
- [31] Lydia T Liu, Horia Mania, and Michael Jordan. Competing bandits in matching markets. In *International Conference on Artificial Intelligence and Statistics*, pages 1618–1628. PMLR, 2020.
- [32] Mohammad Mahdian and Qiqi Yan. Online bipartite matching with random arrivals: an approach based on strongly factor-revealing lps. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 597–606, 2011.
- [33] Vahideh Manshadi and Scott Rodilitz. Online policies for efficient volunteer crowdsourcing. In *Proceedings of the 21st ACM Conference on Economics and Computation*, pages 315–316, 2020.
- [34] Vahideh H Manshadi, Shayan Oveis Gharan, and Amin Saberi. Online stochastic matching: Online actions based on offline statistics. *Mathematics of Operations Research*, 37(4):559–573, 2012.
- [35] Duncan C McElfresh, Christian Kroer, Sergey Pupyrev, Eric Sodomka, Karthik Abinav Sankararaman, Zack Chauvin, Neil Dexter, and John P Dickerson. Matching algorithms for blood donation. In *Proceedings of the 21st ACM Conference on Economics and Computation*, pages 463–464, 2020.
- [36] Aranyak Mehta and Debmalya Panigrahi. Online matching with stochastic rewards. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 728–737. IEEE, 2012.
- [37] Aranyak Mehta, Amin Saberi, Umesh Vazirani, and Vijay Vazirani. Adwords and generalized online matching. *Journal of the ACM (JACM)*, 54(5):22–es, 2007.
- [38] Aranyak Mehta, Bo Waggoner, and Morteza Zadimoghaddam. Online stochastic matching with unequal probabilities. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*, pages 1388–1404. SIAM, 2014.
- [39] Julián Mestre. Greedy in approximation algorithms. In *European Symposium on Algorithms*, pages 528–539. Springer, 2006.
- [40] Marco Molinaro and R Ravi. The query-commit problem. *arXiv preprint arXiv:1110.0990*, 2011.
- [41] Sahil Singla. The price of information in combinatorial optimization. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2523–2532. SIAM, 2018.
- [42] Martin L Weitzman. Optimal search for the best alternative. *Econometrica: Journal of the Econometric Society*, pages 641–654, 1979.