Efficient Nonmyopic Online Allocation of Scarce Reusable Resources

Zehao Dong
Washington University in St. Louis
zehao.dong@wustl.edu

Sanmay Das
George Mason University
sanmay@gmu.edu

Patrick Fowler
Washington University in St. Louis
pjfowler@wustl.edu

Chien-Ju Ho
Washington University in St. Louis
chienju.ho@wustl.edu

ABSTRACT

We study settings where a set of identical, reusable resources must be allocated in an online fashion to arriving agents. Each arriving agent is patient and willing to wait for some period of time to be matched. When matched, each agent occupies a resource for a certain amount of time, and then releases it, gaining some utility from having done so. The goal of the system designer is to maximize overall utility given some prior knowledge of the distribution of arriving agents. We are particularly interested in settings where demand for the resources far outstrips supply, as is typical in the provision of social services, for example homelessness resources. We formulate this problem as online bipartite matching with reusable resources and patient agents. We develop new, efficient nonmyopic algorithms for this class of problems, and compare their performance with that of greedy algorithms in a variety of simulated settings, as well as in a setting calibrated to real-world data on household demand for homelessness services. We find substantial overall welfare benefits to using our nonmyopic algorithms, particularly in more extreme settings – those where agents are unwilling or unable to wait for resources, and where the ratio of resource demand to supply is particularly high.

KEYWORDS

Online Matching; Reinforcement Learning; Randomized Algorithms

ACM Reference Format:
Zehao Dong, Sanmay Das, Patrick Fowler, and Chien-Ju Ho. 2021. Efficient Nonmyopic Online Allocation of Scarce Reusable Resources. In *Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), Online, May 3–7, 2021*, IFAAMAS, 9 pages.

1 INTRODUCTION

Several important problems arising from the need for institutions to allocate scarce societal resources are intrinsically online in nature. For example, when organs from deceased donors become available, they must be quickly matched with recipients on the waiting list [4, 14], and when households experience homelessness (or are at imminent risk of homelessness), they become eligible to receive community-provided homelessness services [5, 13]. In such situations, the institution typically has an allocation rule (often attempting to balance efficiency and equity) that governs who gets the resource [9]. In such situations, the resources are scarce – demand typically far outstrips supply – so it is critical to make reasonable decisions among many eligible recipients.

We focus on online problems where the institution strives for efficient allocation of resources among an eligible population. This defers equity considerations to a prior stage, where eligibility for the resource is determined, and instead focuses on the problem of how to allocate available resources among an eligible population given knowledge of the utility from each match of an agent (e.g. a household experiencing homelessness) to a resource (e.g. space in a shelter). We consider a setting in which there are a number of identical resources. Agents arrive over time; when they arrive, the system becomes aware of the utility of matching that agent with any of the available resources, and the maximum period of time that agents would wait – if that period of time elapses without them being matched, they leave, and the system attains no utility. On the other hand, if they are matched with a resource, the system realizes that utility, and the resource ends up being occupied by that agent (and hence unavailable to others) for a certain period of time. The tradeoff is then between committing a resource into the future versus not (immediately) realizing the available utility.

With the above framing, this work extends standard online bipartite matching to scenarios in which online agents are patient, i.e., agents are willing to wait for some period of time to be matched, and offline resources are reusable, i.e., resources will be released in some period of time after being allocated. These two considerations are practical in nature but relatively under-explored in the online matching literature. Below we summarize our main contributions towards addressing these questions.

- We formulate the problem of online bipartite matching with reusable resources and patient agents (henceforth **OM-RR-PA**).
- We analyze the performance of greedy algorithms for OM-RR-PA and show that greedy algorithms are sub-optimal in situations where (i) resource scarcity is very high, and (ii) agents are unwilling or unable to wait for very long.
- We construct linear programs (LP) for OM-RR-PA under known adversarial distribution [2, 3] that lead to valid upper bounds on the expected offline optimal. We then propose an online algorithm that achieves a competitive ratio of $\frac{1}{2} - \epsilon$ for any ϵ.
- Under the additional assumption that the resource occupation time and agent waiting time are exponentially distributed, we formulate OM-RR-PA as a Markov decision process (MDP). When agents are impatient, we show that the optimal online algorithm

Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), U. Endriss, A. Nowé, F. Dignum, A. Lomuscio (eds.), May 3–7, 2021, Online.

is tractable; When the agents are patient, we propose to utilize reinforcement learning to approximate the optimal policy.

- We evaluate the proposed algorithms both on simulated data and on a real-world dataset that predicts the effects of two different homelessness interventions on future outcomes (return to homelessness) over several years in a major US metro. The experimental results demonstrate that our proposed algorithms perform substantially better than the greedy algorithm and other baselines, especially in regimes with impatient agents and extreme resource scarcity. In addition, our MDP-based algorithms outperform the LP-based algorithms when the exponential distribution assumptions are approximately satisfied.

1.1 Related Work

Online bipartite matching has been extensively studied in the literature, and one promising direction is to formulate the problem using linear programs (LP) and design algorithms accordingly. This approach has been adopted to solve problems in various application domains, including online ad auctions [6, 7], task assignment in crowdsourcing markets [10, 11], and organ transplantation [14]. In these works, online arriving agents are often assumed to be impatient and need to matched upon arrival, and the offline resources are assumed to be disposable (gone when used). However, in many real-world applications, resources might be re-usable and agents might be patient and willing to wait to be matched. Our work differs from the above works by considering these two practical but under-explored aspects in online matching.

One relevant work in this line of research is by Dickerson et al. [8], who consider reusable resource settings for online bipartite matching. They formulate a linear program with novel constraints that generate feasible probabilities of assigning edges at every time step. They develop Monte-Carlo simulation-based online algorithms that use the optimal solution of the proposed linear program. However, in their work, online agents are still assumed to be impatient and need to be matched upon arrival. Our LP-based approaches extend their work to incorporate patient agents.

Since online matching is essentially a sequential decision-making problem, formulating the problem as a Markov decision process (MDP) [17] and solving the optimal policy for online matching is another natural approach. When the environment is complex and exactly solving for the optimal policy is hard, reinforcement learning (RL) [12] is commonly used to approximate the optimal policy. Our MDP-based approaches explore the usage of this approach under certain distributional assumptions. Our formulation shares similarities with work on trade execution problems[15, 16], which formulates the problem as an MDP, with the action space being the limit order prices at which to reposition all remaining inventory, and the state being represented by various statistics of order books. Our formulation is similar in that the wait list in our problem plays a similar role to the order book in the trade execution problem. In the domain of online bipartite matching, Stein et al. [18] apply reinforcement learning approaches to design a matching mechanism that is strategyproof and individually rational for online bipartite matching with reusable resources and impatient agents.

2 SETTING AND PRELIMINARIES

We first formalize the problem of online matching with reusable resources and patient agents (OM-RR-PA). In OM-RR-PA, the policy designer is given as input a bipartite graph $G = (U, V, E)$, where U and V respectively represent the set of offline reusable resources and the set of online agents in the matching system. For each edge $e = (u, v) \in E$, we define weight w_e to denote the utility that could be obtained by matching v and u. We use $N = |U|$ to denote the number of resources and $T = |V|$ to denote the number of arriving agents. At each time step, an agent v from V arrives. A *patient* agent will wait for D_v time steps and leave if not matched. When *reusable* resource u is allocated to agent v, it will be released after K_e ($e = (u, v)$) time steps. We assume that both $\{D_v\}$ and $\{K_e\}$ are bounded. Below we discuss the additional assumptions we make in this paper.

Distributional Assumptions on $\{D_v\}$ and $\{K_e\}$. In this paper, we first discuss the general setting that D_v and K_e can follow any known distributions and propose LP-based methods (Section 3) for this setting. We then consider the scenario in which both the resource occupation time K_e and agent waiting time D_v are exponentially distributed. More formally, K_e and D_v are assumed to be realizations of i.i.d. random variables drawn from exponential distributions with parameters λ_k and λ_d respectively. This corresponds to a natural scenario in which an agent who occupies a resource keeps the resource with a fixed probability every round, and an agent who is waiting for resources keeps waiting with a fixed probability every round. We discuss how we can utilize MDP-based approaches with this assumption (Section 4).

Known Adversarial Distribution (KAD) for Agents. In our setting, the agent distribution is characterized by the utilities $\{w_e\}$. We assume the choice of the distribution could be adversarial but the agents' arrival sequence is stochastic and drawn from the distribution. We denote the PDF and CDF of the utility distribution as f and F. We also assume the distribution is known to the policy designer. This knowledge assumption might be (approximately) satisfied in practice if the designer has access to historical data. KAD is introduced in prior works[2, 3, 8] and is also known as Prophet Inequality matching. For OM-RR-PA with T rounds and an input graph $G = (U, V, E)$, at each time $t \in T$, an agent $v \in V$ is sampled from a known distribution $\{p_{v,t}\}$ such that $\sum_{v \in V} p_{v,t} \leq 1$. Moreover, once we set $p_{v,t} = \frac{1}{|V|}$, the KAD model is equivalent to a KIID (Known IID) input model.

2.1 Analysis of the Greedy Algorithm

We first analyze the performance of the (myopic) greedy algorithm which assigns any available resource to the agent who gains the most immediate utility, without taking into account future arrivals. Such greedy allocation is common. For example, when a space in a homeless shelter becomes available, the agency may offer it to the household ranked as being in the highest need; when deceased donor livers become available, they are offered first to those who are medically matched and with the highest MELD scores, a measure of need.

Theorem 2.1. *In OM-RR-PA with N identical resources, when D_v and K_v are constant such that $D_v = d$, $K_v = k$, and w_e is bounded*

within the range $[L, U]$*, under the worst case agent arrival, the asymptotic competitive ratio of the myopic (greedy) algorithm* CR_{Greedy} *can be characterized as follows:*

$$CR_{Greedy} = \begin{cases} 1 & k \leq N \\ 1 & d \to \infty \\ \frac{U}{L} & k \geq 2N + d \end{cases}$$

The theorem implies that when we have an abundant amount of resources (i.e., the occupation time k for each resource is smaller than the number of resources N, since arrivals are fixed to one per unit time), greedy performs optimally since every agent is getting resources. When agents are patient and are willing to wait for a long period of time (i.e., d is large), greedy also works well. However, when neither of these are true (i.e., we do not have enough resources, and agents can only be allocated resources within a short time frame after arrival), the performance of the greedy algorithm could degrade significantly compared with offline optimal, and thus designing non-myopic online allocation algorithms could bring benefits.

2.2 Overview of Our Approaches

In this paper, we design non-myopic online allocation algorithms. In Section 3, we first extend online bipartite matching problem to settings that combines reusable resources and patient agents. We assume the resource occupation time and agent waiting time are known but can follow any distribution. We formulate the problem as linear programs and develop algorithms accordingly. In Section 4, we consider settings in which the resource occupation time and patient waiting time are exponential distributed. With this assumption, we can formulate the problem as a Markov decision process (MDP) due to the memorylessness property of the exponential distribution. We also discuss the design of online matching algorithms with this formulation.

3 LP-BASED ALGORITHMS FOR OM-RR-PA

In this section, we formulate the linear programming (LP) formulations for OM-RR-PA and discuss the design of online matching algorithms when both resource occupation time and agent waiting time can follow any known distributions. While the focus of this paper is on settings with identical resources, since LP formulations can naturally handle the situation with non-identical resources, in the following discussion, we first discuss the formulation with non-identical resources (denoted by LP-NID) in Section 3.1 and then demonstrate how to design more efficient algorithms for the formulation with identical resources (denoted by LP-ID) in Section 3.2.

3.1 OM-RR-PA with Non-Identical Resources

LP-NID Formulation. Let a bipartite graph $G = (U, V, E)$ be the input to OM-RR-PA with N non-identical resources. Suppose that the online matching problem has a horizon of T, and $\{D_v\}$ is upper bounded by d. In this formulation, both D_v and K_e are random variables with known distribution, and E_v (E_u) denotes the set of edges incident to agent vertex v (resource vertex u). For a potential assignment $e = (u, v)$, we use variable $x_{e,t,n}$ to denote the assignment decision, where $x_{e,t,n}$ represents the probability that (agent) v arrives at time t and is assigned to (resource) u at n steps after the arrival. For notation simplicity, we set $D_e = D_v$ if $e = (u, v)$, since we assume the waiting time of online agent is irrelevant to offline resources. LP-NID can then be formulated as follows. Note that our formulation of LP-NID and the corresponding algorithm design introduced later is an extension of the work by Dickerson et al. [8] to include the consideration of patient agents while they only consider impatient agents.

$$\max \sum_{e\in E}\sum_{t\in T}\sum_{n=0}^{d} \Pr(D_e \geq n) x_{e,t,n} w_e \tag{1}$$

$$\text{s.t.} \sum_{e\in E_v}\sum_{n=0}^{d} x_{e,t,n} \leq p_{v,t} \quad \forall (t, v) \tag{2}$$

$$\sum_{t'\leq t}\sum_{e\in E_u}\sum_{n=0}^{\min(d,t-t')} x_{e,t',n} \Pr(D_e \geq n) \Pr(K_e \geq t-t'-n) \leq 1 \quad \forall (t, u) \tag{3}$$

$$0 \leq x_{e,t,n} \leq 1 \quad \forall (e, t, u) \tag{4}$$

Corollary 3.1. *The optimal value of LP-NID provides an upper bound on the expected overall utility of OM-RR-PA with non-identical resources.*

Claim 3.2. *Linear program for online matching with reusable resources and impatient agents (as discussed in Dickerson et al. [8]) is a special case of LP-NID by setting the upper bound d to be 0.*

Let us interpret the above linear program. First of all, for the objective, let $\Omega_{e,t}$ be the event that assignment $e = (v, u)$ gets assigned and agent v arrives at time t. The conditional probability of $\Omega_{e,t}$ under the condition that the occupation time is D_e can be computed as: $Pr(\Omega_{e,t}|D_e) = \sum_{n=0}^{d} x_{e,t,n} I(n \leq D_e)$, where $I(*)$ is the indicator function. Let $f_D(*)$ be the PDF of random variable D_e, then we get the unconditional probability of $\Omega_{e,t}$ such that $Pr(\Omega_{e,t}) = \int \sum_{n=0}^{d} x_{e,t,n} I(n \leq D_e) f_D(D_e)\, dD_e$. Therefore, $Pr(\Omega_{e,t}) = \sum_{n=0}^{d} Pr(D_e \geq n) x_{e,t,n}$, leading to the expected matching utility as computed in the objective. Constraint (2) guarantees the probability of assigning v arrives at time t be no larger than the the probability that v arrives at t in all cases. Constraint (3) guarantees the probability that resource u is used up at time t to be smaller than 1. The formulation extends the one in previous work [8] by incorporating patient agents. In particular, we use the law of total expectation to incorporate patient agents in the objective as well as constraints that generate feasible probabilities of the potential assignments.

LP-Based Online Algorithm. We design online adaptive algorithm, Algorithm 1: OAA-NID(ϕ), using optimal solutions $\{x^*_{e,t,n}\}$ of LP-NID and Monte-Carlo simulations. Let $\alpha_{e,t}$ be the probability that assignment e is available at time t. As discussed in prior work [1, 8], $\alpha_{e,t}$ could be approximated with arbitrarily small error. Under the condition that v arrives at time t with a waiting time of D_v and e is available at $t \in \Sigma$, where $\Sigma \subseteq \{t, t+1, .., t+D_v\}$, the conditional probability that OAA-NID(ϕ) assigns edge $e = (u, v)$ at time $t+n \in \Sigma$ is $\frac{\phi x^*_{e,t,n}}{\alpha_{e,t+n} p_{v,t}} I(n < D_v)$, leading to an unconditional probability of $\alpha_{e,t+n} p_{v,t} \int \frac{\phi x^*_{e,t,n}}{\alpha_{e,t+n} p_{v,t}} I(n < D_v) f_D(D_v)\, dD_v = \phi x^*_{e,t,n} \Pr(D_e >$

n). As an extension to the prior result [8], OAA-NID(ϕ) achieves a competitive ratio of ϕ once $\phi \le \alpha_{e,t}$ for any e and t.

Algorithm 1 OAA-NID(ϕ): Online Adaptive Algorithm for Non-Identical Resources

For each time t, let v and PAD denote the agent arriving at time t and a set of previous allocation decisions.
Choose n and $e = (u, v)$ such that $n \le D_v$ and $n \in \Sigma$ with probability $\frac{\phi x^*_{e,t,n}}{\alpha_{e,t+n} p_{v,t}}$, and add allocation decision $(e, t+n)$ into PAD.
for allocation decision (e', t') in PAD **do**
  if e' is free at time t and $t' = t$ **then**
    Match e' at time t
  end if
end for

Theorem 3.3. *In OM-RR-PA with non-identical resources, OAA-NID(ϕ) achieves a competitive ratio of $\frac{1}{2} - \epsilon$ for any $\epsilon > 0$.*

3.2 OM-RR-PA with Identical Resources

We now discuss the setting with identical resources. While LP-NID and the corresponding algorithms can still be applied, we can adjust LP-NID to a more efficient linear program (LP-ID) for OM-RR-PA with identical resources.

LP-ID Formulation and Corresponding Online Algorithm. We use variable $x_{v,t,n}$ to denote the assignment decision, i.e., it represent the probability that agent v arriving at time t is matched n time steps after the arrival (since the resources are identical, we do not need to index the resources.) The corresponding linear program is formulated as LP-ID. Compared with LP-NID, LP-ID has less variables and constraints and is therefore more computationally efficient. As previously discussed, we could design Algorithm 2 OAA-ID(ϕ) based on simulation results and optimal solution of LP-ID. In OAA-ID(ϕ), α_t represents the probability that there are free resources at time t, and ϕ is still required to be smaller than α_t.

$$\max \sum_{v \in V} \sum_{t \in T} \sum_{n=0}^{d} \Pr(D_v \ge n) x_{v,t,n} w_v \tag{5}$$

$$s.t. \sum_{n=0}^{d} x_{v,t,n} \le p_{v,t} \quad \forall (t, v) \tag{6}$$

$$\sum_{t' \le t} \sum_{v \in V} \sum_{n=0}^{\min(d, t-t')} x_{v,t',n} \Pr(D_v \ge n) \Pr(K_v \ge t - t' - n) \le N \quad \forall t \tag{7}$$

$$0 \le x_{v,t,n} \le 1 \tag{8}$$

When agents are impatient, the linear program is a special case of LP-ID with $d = 0$, thus variables $x_{v,t,n}$ degenerate to $x_{v,t}$, and all $Pr(D_v \ge n) = 1$. In the corresponding online algorithm, we do not need to consider the set of previous allocation decisions, and the conditional probability that v gets assigned is $\frac{\phi x^*_{v,t}}{\alpha_t p_{v,t}}$.

Algorithm 2 OAA-ID(ϕ): Online Adaptive Algorithm for Identical Resources

For each time t, let v and PAD denote the agent arriving at time t and a set of previous allocation decisions.
Choose n such that $n \le D_v$ and $n \in \Sigma$ with probability $\frac{\phi x^*_{v,t,n}}{\alpha_{t+n} p_{v,t}}$, and add allocation decision $(v, t+n)$ into PAD.
for allocation decision (v', t') in PAD **do**
  if There are free resources and $t' = t$ **then**
    Match v' with an arbitrary free resource at time t
  end if
end for

4 MDP-BASED ALGORITHMS FOR OM-RR-PA UNDER EXPONENTIAL ASSUMPTION

So far we have introduced LP-based methods in settings where we do not make distributional assumptions about agent waiting time and resource occupation time. In this section, we explore settings where we assume these are exponentially distributed and introduce MDP-based algorithms. Recall that under these assumptions, agent waiting time D_v and resource occupation time K_v are exponentially distributed such that $D_v \sim Expo(\lambda_d)$ and $K_v \sim Expo(\lambda_k)$. These assumptions align well with many applications, as we demonstrate in analyzing our real-world dataset in Section 5.

4.1 Online Matching with Impatient Agents

We first address a simpler scenario in which agents are impatient and need to be matched immediately upon arrival. To design an MDP-based policy, we need to decide on the state representation of the online matching system and action space in which the policy designer searches for the optimal matching policy. We also need to formulate the corresponding reward and state transition functions.

- State $s = (n, t)$: Each state s can be represented by a pair (n, t), where $n \in \{0, 1, ..., N\}$ is the number of resources that are occupied and $t \in \{1, ..., T\}$ is the time round. The initial state of the system is $s_1 = (n_1, t_1) = (0, 1)$.
- Action space: We consider an action to be represented by choosing a threshold, i.e., an agent is assigned a resource if and only if the utility for obtaining the resource is higher than the threshold. We denote the threshold space as Θ, the continuous input space of the known matching utility distribution.
- Rewards $R((n, t), a)$: The immediate reward the system obtains by taking action a at state (n, t). Recall that f is the PDF of the utility distribution. Therefore, we have $R((n, t), a) = \int_a^\infty x f(x)\, dx$.
- State transition $T((n', t')|(n, t), a)$: The probability of transitioning to state (n', t') by taking action a in state (n, t). Note that t is increasing by 1 after each action. Therefore, we can focus on the transition on n. For notational simplicity, let $B(n, n')$ denote the probability that, out of n resources, n' of them are still occupied at the next time step. Since resource occupation follows an exponential distribution, $B(n, n')$ is easy to compute. We also let $B(n, n') = 0$ for invalid choices of (n, n'): they are invalid when $n' > n$ or when $n, n' \notin \{0, ..., N\}$. Recall that F is the

CDF of the utility distribution. Therefore $F(a)$ is the probability that an arriving agent is not allocated a resource when the threshold is a. The probability of transitioning to a state with n' is the sum of the probability of allocation and n' still occupied (i.e, $(1-F(a))B(n+1,n')$) and the probability of not allocating and having n' resources still occupied (i.e., $F(a)B(n,n')$), thus $T((n',t')|(n,t),a)$ is

$$\begin{cases} F(a)B(n,n') + (1-F(a))B(n+1,n') & n \leq N-1,\ t'=t+1 \\ B(n,n') & n=N,\ t'=t+1 \\ 0 & \text{otherwise} \end{cases}$$

The goal of the system designer is to find a policy $\pi(n,t)$ that determines a threshold for each state that maximizes the total expected reward over T rounds. Let s_t be the state at time t assuming the system follows policy π. The system's reward for following the policy π starting at state s_1 can be written as $U(\pi, s_1) = \sum_{t=1}^{T} R(s_t, \pi(s_t))$.

Note that with this finite-horizon MDP formulation, the optimal policy is efficiently solvable using a standard backprojection algorithm, as introduced next.

Backprojection Algorithm. This MDP can be solved exactly using backprojection, a dynamic-programming algorithm. Let $A_{n,t}$ denote the maximum expected total utility when n resources are occupied at time t, and $a_{n,t}$ denote the corresponding optimal threshold to choose as the action. Below we describe how to derive the optimal policy by setting these two values in a backward manner.

First consider the boundary condition in the last round (i.e., $t=T$). When there are remaining resources (i.e., $n<N$), we can set $a_{n,t}=0$ (assign resources without conditions) and $A_{n,t} = (N-n)\int_0^\infty x f(x)\,dx$. When there is no available resource (i.e., $n=N$), we set $a_{n,t}=\infty$ (no resource to allocate) and $A_{n,t}=0$.

For round $t<T$, given the knowledge of $A_{n,t+1}$ and $a_{n,t+1}$ for all n, we can calculate $A_{n,t}$ and $a_{n,t}$ using standard dynamic programming approaches. We can then obtain the optimal policy through backpropagation from $t=T$ to 0.

4.2 Online Matching with Patient Agents

We now consider the more general, complex online matching in which agents might be willing to wait for some number of rounds. In this scenario, the system's decision could depend on the waitlist, i.e., the list of agents who are waiting to be allocated resources, in addition to the number of resources. Therefore, the state representation needs to take the waitlist into account. Since agents are heterogeneous (obtaining different utility when being allocated resources), the state representation is more complicated. As such, we develop a reinforcement learning algorithm to approximate the optimal threshold policy.

- State $s = (w_1 ... w_N, r, t)$: $w_1,..w_N$ are the largest N matching utilities in the waiting list such that $w_1 \geq w_2 .. \geq > w_N$. When the size of the waiting list (denoted as h) is smaller than N, the last $N-h$ of these values are set to be 0; r is the number of used resources and t is the current time step. The agent waiting times D_v are exponentially distributed, thus only the largest N matching utilities in the waiting list should influence the selection of threshold due to the memorylessness of the distribution.
- Action space: We again adopt the threshold policy, i.e., the action is to select a threshold to match agents in the waiting list whose utility is larger than the selected threshold. In our proposed algorithm, we use a discrete threshold space of size M. When the matching utility distribution has an upper bound H_u and a lower bound H_l, the discrete threshold space can be formulated as $\{H_l + \frac{i-1}{M}(H_u - H_l) | i = 0, 1, 2..M-1\}$. When the matching utility is unbounded, suppose $F(*)$ is the CDF of the utility distribution, then the threshold space is formulated as $\{F^{-1}(\frac{i-1}{M}) | i = 0, 1, 2..M-1\}$.

The reward and the state transition can then be written down accordingly based on the above state and action representations. Note that given the large state space, this MDP is challenging to solve exactly. Therefore, we propose BQL: Backprojected Q-values Learning Algorithm, which utilizes reinforcement learning to approximate the Q-values for all states, which in turn provides an approximately optimal policy. The BQL algorithm follows a similar scheme to the RL algorithm for optimal trade execution of Nevmyvaka et al. [15]. In particular, we first conduct N simulations where the actions are randomly selected to get N training waiting list sequences. We then train the deep Q network based on N waiting list samples at each time round t (from T to 0). The returned Q values can then be used as a representation of the (approximately) optimal policy.

Algorithm 3 BQL: Backprojected Q-values Learning Algorithm

```
for t = T to 0 do
    Current waiting list → w_1...w_N
    for r = 0 to m do
        State s = (w_1...w_N, r, t)
        for i = 0 to M-1 do
            Compute thresholds: a = H_l + ((i-1)/M)(H_u - H_l)
            Compute reward r based on a and s
            Simulate state transition s → s'
            if t = T then
                Update q value for (s, a): r(s, a)
            else
                Update q value for (s, a): r + max(q(s'))
            end if
        end for
    end for
    Fit deep neural network q
end for
```

5 EXPERIMENTS

In this section, we evaluate the effectiveness of the proposed LP-based algorithms and MDP-based algorithms in online matching problems on both simulations and a real-world dataset. In these experiments, the arrival model is set as KIID (Known-IID), and resources are set to be identical. All experimental results are based on 10,000 evaluation runs.

5.1 Simulations: Impatient Agents

Experimental Setting: We first examine algorithms for OM-RR-PA with impatient agents using simulations. We set the matching

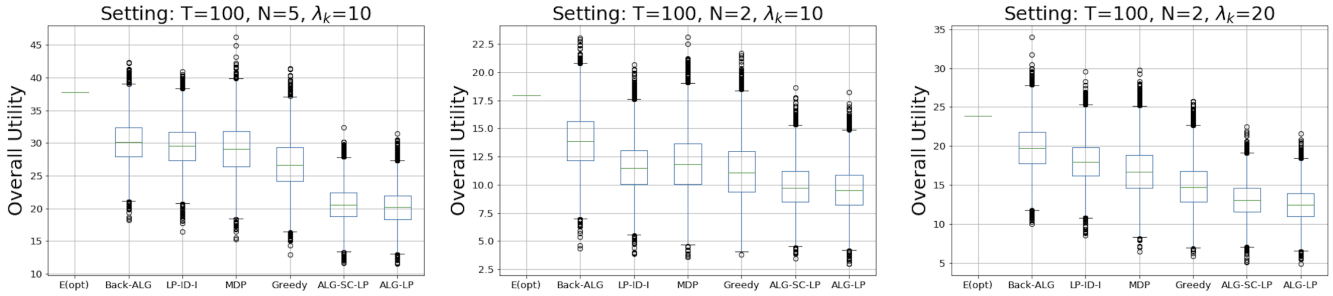

Figure 1: OM-RR-PA with identical resources and impatient agents: Performance comparison of proposed algorithms: LP-based algorithm (LP-ID-I), MDP-based algorithm (Back-ALG), greedy baselines and other state-of-art LP-based algorithms (ALG-SC-LP, ALG-LP)

utility to be uniformly distributed between 0 and 1. We conduct the experiments in settings with different number of resources N and average resource occupation time λ_k. Inspired by the experimental setup in prior work [8, 19], we test the following six algorithms. (1) Back-ALG is an MDP-based algorithm with the backprojection approach proposed in subsection 4.1; (2) LP-ID-I is the LP-based algorithm for impatient agents in subsection 3.2. We make the small modification that an agent v that arrives at t is matched with a probability of $\frac{x^*_{v,t}}{p_{v,t}}$ when there are free resources at time t. [1] (3) The MDP approach formulates an infinite-horizon MDP with the same state formulation, reward and transition function as subsection 4.1, then solves the MDP to get the assignment rule for each state and applies it in the online version. (4) ALG-SC-LP and ALG-LP are LP-based algorithms in [8]. (5) The greedy algorithm assigns available resources to agents who gain the most immediate utility. (6) E(opt) is a valid upper bound on the expected overall utility (see subsection 3.2).

Results: Figure 1 shows that the proposed threshold-based algorithm Back-ALG significantly outperforms other algorithms in all settings where agents are impatient. In addition, the improvement over the greedy algorithm is more substantial when the resource is scarce (comparing the middle graph and the left graph in Figure 1) and the resource occupation time is relatively longer (comparing the middle graph and the right graph in Figure 1). These observations are consistent with Theorem 2.1. In addition, when agents are impatient, our proposed LP-based algorithm LP-ID-I always beats the greedy algorithm and outperforms other state-of-the-art LP-based algorithms (ALG-LP,ALG-SC-LP).

5.2 Simulations: Patient Agents

Experimental Setting: Now we evaluate the performance of the proposed algorithms in OM-RR-PA with patient agents. The average resource occupation time is set to be 20 (i.e., $\lambda_k = 20$), and the average agent waiting time λ_k is selected from the set $\{1, 2, 4, 8\}$. $\frac{\lambda_k}{\lambda_d}$ reflects the ratio of resource demand to supply. The matching utility distribution is selected from the Beta distribution family. We test four algorithms: (1) LP-ID-P is the LP-based algorithm for patient agents in subsection 3.2. We use the modification as LP-ID-I, thus the allocation decision that agent v arrives at time t is matched after n time steps is sampled with probability $\frac{x^*_{v,t,n}}{p_{v,t}} I(n \le D_v)$, where $I(*)$ is the indicator function. In addition, we assume that the upper bound of waiting time is 15 ($d = 15$), since the exponential distribution is unbounded and setting maximum potential waiting time d to be T makes solving LP-ID time-consuming. (2) The optimal solution of LP-ID. Base on the above assumption, the optimal value of LP-ID is also an approximated value. (3) The BQL algorithm that first performs exploration under the random assignment rule and collects samples (sequences of length N), then trains a deep Q network as in Algorithm 3. (4) The greedy algorithm.

Results: Figures 2 and 3 demonstrate that BQL always outperforms the greedy algorithm. Moreover, Figure 3 shows that the relative improvement of BQL over greedy is more substantial when resources are scarce and the ratio of resource demand to supply is high. This again aligns with our theoretical analysis indicating that these are the most difficult conditions for greedy, and therefore our algorithm has more room to improve.

In addition, Figure 2 show that BQL attains higher average overall utilities than LP-ID-P (e.g. SubGraph4 and SubGraph7) in most settings.Though the LP-based algorithm LP-ID-P does not use information from the distribution of occupation time and of waiting time, it nevertheless improves substantially over the greedy algorithm. In addition, there are occasions when LP-ID-P beats BQL (e.g. SubGraph1). Since the LP-based algorithms are more computationally efficient and could be applied in more general settings, they could have great potential in real-world applications.

5.3 Real-World Dataset Experiment

Finally, we evaluate the LP-based algorithm LP-ID-P and the MDP-based algorithm BQL on a real-world dataset for homelessness services. This dataset includes estimated re-entry probabilities for four different interventions that could be given to homeless households in a major US metro [13]. These were all households that were eligible to receive services, but received different levels of interventions. We focus on two of the interventions, transitional housing (the most intensive one) and emergency shelter, with the idea being that transitional housing is the scarce resource, and agents who do not receive it can potentially wait in emergency shelters.

[1] This modification is for computational efficiency. In practice, the results are similar without this modification, observed in both our own experiments and prior work [8]

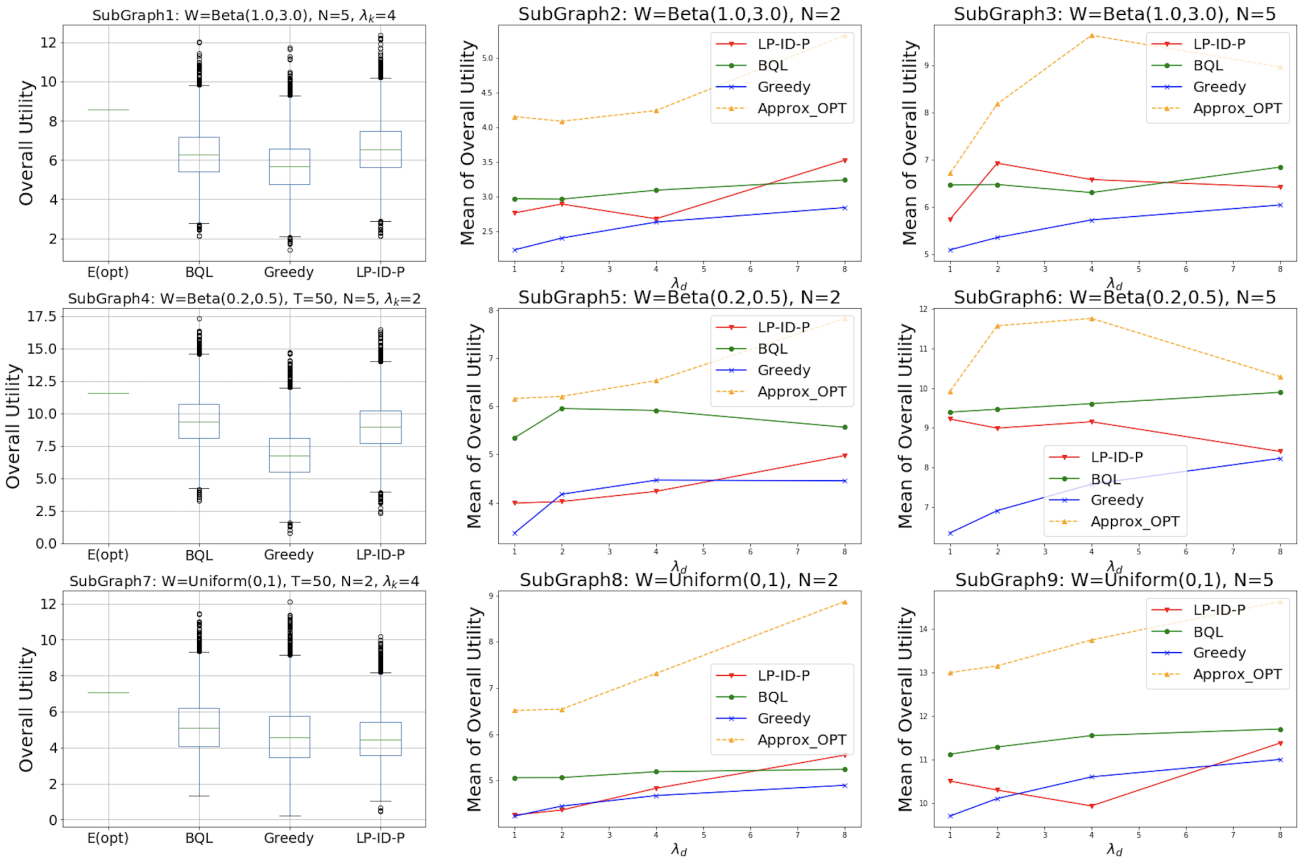

Figure 2: OM-RR-PA with identical reusable resources and patient agents: Performance comparison of proposed LP-based algorithm (LP-ID-P), proposed MDP-based algorithm (BQL) and greedy baseline

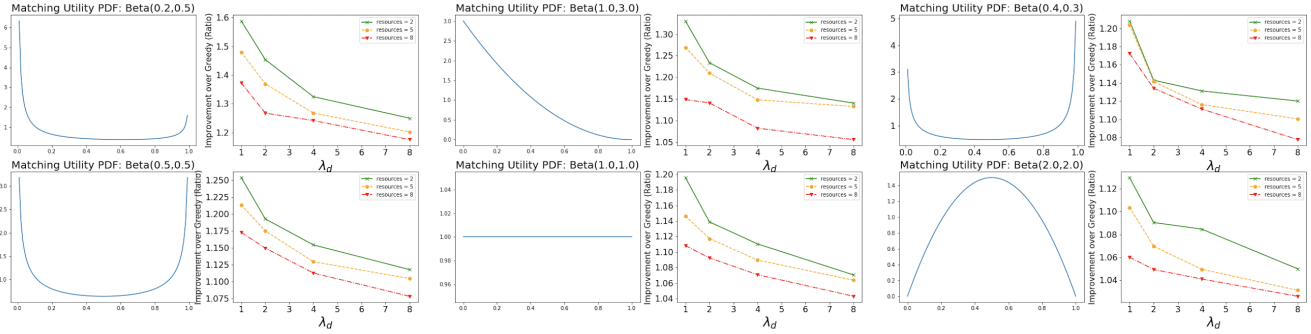

Figure 3: OM-RR-PA with identical reusable resources and patient agents: Relative performance improvement of BQL algorithm over greedy algorithm under different settings

Experimental Setting: Our experiment setting is calibrated using the real dataset. We use the difference in estimated re-entry probabilities between the two as our measure of utility (that is, the utility is the decrease in the probability that a household would become homeless again in the next two years if they were given transitional housing instead of emergency shelter). We also calibrate the mean time spent in transitional housing using the real dataset. Transitional housing (TH) is relatively scarce in the data, with less than 20% of households receiving that intervention. The objective is to maximize the utility over a half year (i.e. $T = 180$).

In addition to greedy and offline optimal, below we describe the algorithms used in the experiments. (1) In BQL, we use a truncated gamma distribution Gamma(1.0,0.2) to simulate the matching utility during the training phase, while the occupation times for resources are generated by an exponential distribution whose mean is the same as the average over time spent in TH. Figure 4 measures

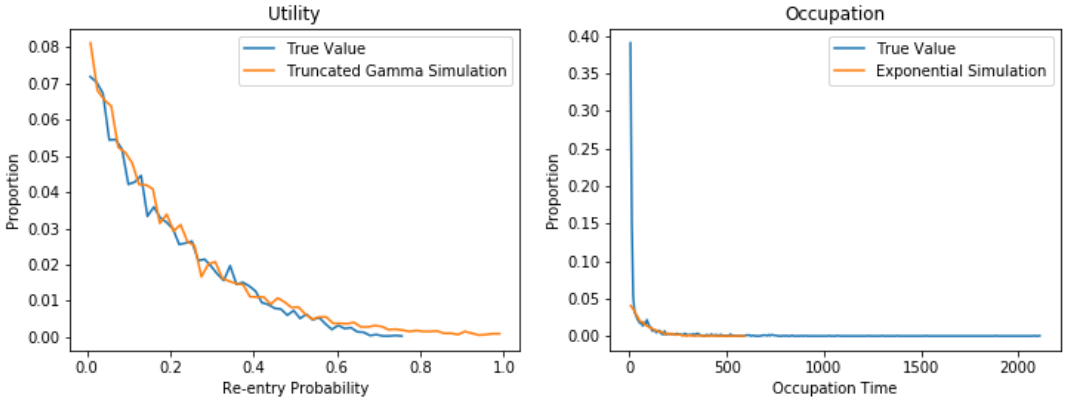

Figure 4: Performance of simulators for re-entry probability (matching utility) and occupation time

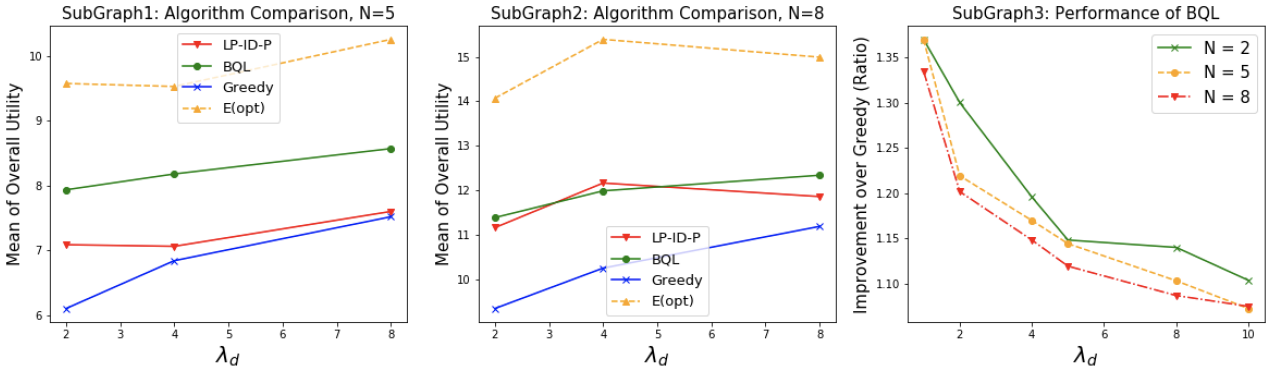

Figure 5: Real-world homelessness dataset evaluation: Comparison of proposed LP-based algorithm (LP-ID-P), proposed MDP-based algorithm (BQL) and the greedy baseline as a function of the agent patience parameter

the performance of these simulators. In the testing period, both matching utilities and resource occupation times are sampled from the dataset. (2) In LP-ID-P, the edge weights w_e of the input graph $G = (V, U, E)$ and resource occupation time are sampled from the dataset. We assume the bound on agent waiting time is 20.

Results: Figure 5 shows our main results. First, in the left graph (SubGraph1) and middle graph (Subgraph2), we compare LP-ID-P and BQL with the greedy algorithm in significantly different settings. Both LP-ID-P and BQL substantially outperform the greedy algorithm. In addition, Subgraph 2 shows that the performance of LP-ID-P is competitive compared with the RL algorithm BQL in some settings. The fact that LP-based algorithms are more computationally efficient, combined with the exponential distribution assumption being approximately satisfied in many real-world settings, make the case that LP-ID-P could be quite powerful in real-world applications.

Second, the right graph (SubGraph3) presents the relative performance improvement of BQL over greedy as a function of agent patience for 3 different possible values of the number of resources available. It demonstrates BQL clearly outperforming the greedy algorithm, bringing up to 35% more benefit, especially in the regime when agents are impatient.

6 CONCLUSION

We study online bipartite matching problems with reusable resources and patient agents. We theoretically characterize regimes where greedy allocation mechanisms may not be efficient – typically when agents in the allocation system are impatient and resource scarcity is high. We develop online algorithms for performance improvement using two different techniques - formulating the problems as linear programs and as Markov decision processes (MDPs). In the former, we extend prior work to the case of online patient agents and propose LP-based algorithms with theoretical performance guarantees. In the latter, with additional distributional assumptions about resource occupation time and agent waiting time, we develop an MDP formulation and algorithms for solving the policy for online matching. Experimental results, based on a variety of simulated settings as well as a setting calibrated to real-world data, demonstrate that our algorithms outperform baseline methods and significantly improve upon the greedy algorithm in regimes with impatient agents and scarce resources.

ACKNOWLEDGMENTS

We are grateful for support from NSF awards 1910392 and 1927422 and a WUSTL OVCR Seed Grant for Interdisciplinary Research.

REFERENCES

[1] Marek Adamczyk, Fabrizio Grandoni, and Joydeep Mukherjee. Improved approximation algorithms for stochastic matching. In *Algorithms-ESA 2015*, pages 1–12. Springer, 2015.

[2] Saeed Alaei, MohammadTaghi Hajiaghayi, and Vahid Liaghat. Online prophet-inequality matching with applications to ad allocation. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, pages 18–35, 2012.

[3] Saeed Alaei, MohammadTaghi Hajiaghayi, and Vahid Liaghat. The online stochastic generalized assignment problem. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 11–25. Springer, 2013.

[4] Ross Anderson, Itai Ashlagi, David Gamarnik, and Yash Kanoria. Efficient dynamic barter exchange. *Operations Research*, 65(6):1446–1459, 2017.

[5] Molly Brown, Camilla Cummings, Jennifer Lyons, Andrés Carrión, and Dennis P Watson. Reliability and validity of the Vulnerability Index-Service Prioritization Decision Assistance Tool (VI-SPDAT) in real-world implementation. *Journal of Social Distress and the Homeless*, 27(2):110–117, 2018.

[6] Niv Buchbinder, Kamal Jain, and Joseph Seffi Naor. Online primal-dual algorithms for maximizing ad-auctions revenue. In *European Symposium on Algorithms*, pages 253–264. Springer, 2007.

[7] 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.

[8] John P Dickerson, Karthik A Sankararaman, Aravind Srinivasan, and Pan Xu. Allocation problems in ride-sharing platforms: Online matching with offline reusable resources. In *Thirty-Second AAAI Conference on Artificial Intelligence*, pages 1007–1014, 2018.

[9] Jon Elster. *Local Justice: How Institutions Allocate Scarce Goods and Necessary Burdens*. Russell Sage Foundation, 1992.

[10] Chien-Ju Ho and Jennifer Wortman Vaughan. Online task assignment in crowdsourcing markets. In *AAAI Conference on Artificial Intelligence*, pages 45–51, 2012.

[11] Chien-Ju Ho, Shahin Jabbari, and Jennifer Wortman Vaughan. Adaptive task assignment for crowdsourced classification. In *Thirtieth International Conference on Machine Learning*, pages 534–542, 2013.

[12] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.

[13] Amanda Kube, Sanmay Das, and Patrick J. Fowler. Allocating interventions based on predicted outcomes: A case study on homelessness services. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 622–629, 2019.

[14] Zhuoshu Li, Kelsey Lieberman, William Macke, Sofia Carrillo, Chien-Ju Ho, Jason Wellen, and Sanmay Das. Incorporating compatible pairs in kidney exchange: A dynamic weighted matching model. In *Proceedings of the ACM Conference on Economics and Computation*, pages 349–367, 2019.

[15] Yuriy Nevmyvaka, Yi Feng, and Michael Kearns. Reinforcement learning for optimized trade execution. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 673–680, 2006.

[16] Brian Ning, Franco Ho Ting Ling, and Sebastian Jaimungal. Double deep q-learning for optimal execution. *arXiv preprint arXiv:1812.06600*, 2018.

[17] Martin L Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 1994.

[18] Sebastian Stein, Mateusz Ochal, Ioana-Adriana Moisoiu, Enrico Gerding, Raghu Ganti, Ting He, and Tom La Porta. Strategyproof reinforcement learning for online resource allocation. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems*, pages 1296–1304, 2020.

[19] Yongxin Tong, Jieying She, Bolin Ding, Libin Wang, and Lei Chen. Online mobile micro-task allocation in spatial crowdsourcing. In *32nd IEEE International Conference on Data Engineering (ICDE)*, pages 49–60. IEEE, 2016.