Bandit Policies for Online Task Assignment with Reuseable Resources

SUBMISSION XXX

We study online task assignment with reusable resources, a problem arising in many practical settings including paper-reviewer assignment for peer review, rideshare driver-rider matching, patient-doctor matching, skill-based call routing, and crowdsourcing platforms. At each time step, a task arrives and must be immediately assigned to k resources from a pool of R reusable options. Each assignment yields a context-dependent reward but makes the assigned resource unavailable for d subsequent time steps. The goal is to maximize long run average reward.

We formulate the problem as a contextual restless multi-armed bandit (CRMAB) and develop an occupancy-measure-based LP which upper bounds the optimal reward. Starting with this LP formulation, we make two contributions. First, we develop a dual Lagrangian index policy and prove it is asymptotically optimal in a mean-field scaling regime where the numbers of resources and arrivals per timestep grow. Second, we relax the LP and prove indexability of the underlying bandit to derive a scalable closed-form Whittle index policy.

Finally, we evaluate our algorithms using trace-based simulation from peer-review datasets and ridesharing benchmarks. The Lagrangian index achieves 94–99% of the LP upper bound across all load levels. Both the Lagrangian and Whittle index policies significantly outperform the natural greedy assignment scheme, improving reward by up to 20% on journal data and 66% on structured synthetic instances.

Contents

Manuscript submitted for review to the 27th ACM Conference on Economics & Computation (EC'26).

1 Introduction

Leading peer review journals routinely process hundreds of submissions per month. Editors must ensure that each paper is peer-reviewed by well-matched experts, while preventing reviewer overload and burnout. In practice, most journals rely on manual assignment, with editors accounting for conflicts of interest and drawing on informal knowledge of reviewer expertise [Biswas and Hasan, 2007]. Even when automation is used, reviewer assignment is often performed greedily on a paper-by-paper basis, assigning each paper to the currently available reviewers with the most suitable expertise. Such a greedy assignment does not for reviewers' expertise profiles and potential future reviewer needs [Fang and Zhai, 2007, Zhao and Zhang, 2022]. Given the rapid growth in submission volumes, more principled reviewer assignment policies are needed.

Similar online assignment problems arise in many settings beyond peer review. In healthcare, patients arrive at clinics and must be matched to available physicians with relevant medical expertise. In customer support systems, incoming tickets must be routed to agents with appropriate skills. In emergency dispatch settings, units must be dispatched to calls based on proximity and capability. In a data center, arriving jobs have different affinity for different servers, and must be routed in a globally efficient manner.

In each of these settings, tasks arrive sequentially over time and must be immediately assigned to available resources. Arriving tasks have some match quality with each of the resources. The match quality could be expertise similarity-based, as in the case of paper-reviewer, patient-doctor and ticket-agent matching. It may be proximity based, as in the case of emergency dispatch and ridesharing. The match-quality may also be a binary compatibility graph, as in the case of eligibility criteria in crowd-sourcing tasks or job-server compatibility in cloud systems scheduling. In all these cases, the resources are *reusable* as well. Once assigned, a resource becomes unavailable for some service time. The *objective* is to maximize the sum of match quality scores obtained over a long run of arriving tasks. In all cases, the decision-maker faces a common tradeoff: balancing the value of the current match against the future availability of the assigned resource.

1.1 Our Model

Suppose we are given a pool of R reusable resources and a stream of arriving tasks. At each time $t \in \mathbb{N}$, a task of type $v_t \in [V]$ arrives, where v_t is drawn i.i.d. with $\Pr[v_t = v] = p_v$. We are given a similarity score matrix $S \in \mathbb{R}^{V \times R}$, where $s_{vr} \in \mathbb{R}$ is the match quality between task type $v \in [V]$ and resource $r \in [R]$. We treat S as known and deterministic.

In the reviewer-assignment setting, tasks correspond to submitted papers and resources correspond to reviewers; the reward s_r represents the suitability or expertise match of reviewer r for paper of type v_t.

Upon arrival of task v_t, we must immediately and irrevocably assign k resources from the currently available resource pool. If the set $\mathcal{A}_t \subseteq [R]$ of resources is assigned to task v_t, the system accrues reward

$$\text{Reward} = \sum_{r \in \mathcal{A}_t} s_{v_t, r}.$$

Each assigned resource $r \in \mathcal{A}_t$ then becomes *busy* for a fixed service duration of d time steps, and is unavailable for assignment to new tasks during times $t+1, t+2, \ldots, t+d$. The objective is to maximize the average score per task over a long time horizon obtained by online assignment policies. Namely,

$$\text{maximize} \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{r \in \mathcal{A}_t} s_{v_t, r}. \tag{1}$$

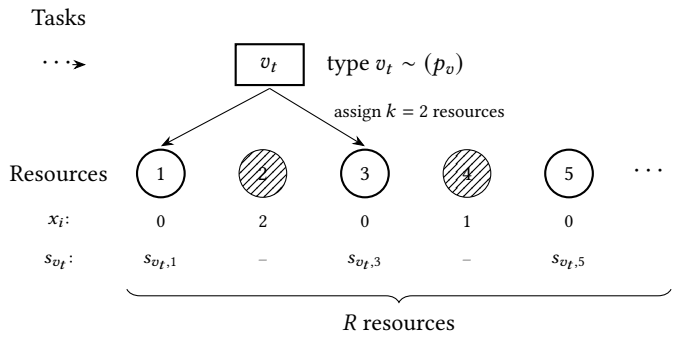


Fig. 1. Online assignment with reusable resources. At each time t, a task of type $v_t \in [V]$ (drawn i.i.d. with $\Pr[v_t = v] = p_v$) arrives and is assigned to k available resources. State $x_i \in \{0, \ldots, d\}$ denotes time steps until resource i is free; $x_i = 0$ means available (shown: $k = 2$). Hatched circles indicate busy resources.

1.2 Existing approaches to this problem

We briefly survey four natural approaches to this problem: online greedy assignment, dynamic programming, offline integer programming, and LP-guided online algorithms. Each offers a different tradeoff between computational tractability, solution quality, and applicability to the online setting.

Online Greedy Assignment. A natural baseline for this problem is a *greedy assignment policy*, which assigns each arriving task to the k available resources with the largest current match rewards. Variants of this approach are used in some automated online assignment systems [Fang and Zhai, 2007, Zhao and Zhang, 2022]. However, greedy assignment fails to account for future demand on different resources. In particular, some resources may be moderately suitable across many task types, while others are specialists with high suitability in a narrow range. Greedy policies may overuse broadly suitable resources (generalists) and underutilize specialists, even when it would be preferable to reserve generalists for future arrivals. Similarly, resources with high suitability for high-demand task types may need to be preserved, even if they appear to be good matches for lower-demand tasks.

Dynamic Programming. Our expected reward maximization problem can be formulated as a Markov decision process (MDP). The system state can be modeled by the vector of remaining busy times for each resource. The optimal policy may then be solved via dynamic programming. However, this formulation is computationally intractable for realistic problem sizes: with R resources and service duration d, the state space grows as $(d + 1)^R$, which is exponential in the number of resources.

Offline Integer Programming. At the other extreme, if all future arrivals were known in advance, the problem could be formulated as a large integer linear program, with binary variables x_{tr} indicating whether task t is assigned to resource r. However, future arrivals are unknown, and repeatedly solving such large optimization problems is computationally prohibitive.

Linear programming guided online algorithms. Recent work has studied related online assignment problems with reusable resources. In particular, our model is closely related to the OM-RR-KIID (Online Matching with Reuseable Resources and Known i.i.d. distribution of arrivals) framework introduced by [Dickerson et al., 2021, Nanda et al., 2020, Sumita et al., 2022] for rideshare rider–driver matching. Their setting focuses on one-to-one matching, whereas we consider one-to-many task assignment. Most existing algorithms and heuristic policies in the literature rely on solving linear

programs, which is computationally feasible in problem scales with few (10-100) available resources (see [Dickerson et al., 2021], [Sumita et al., 2022], [Nanda et al., 2020]). However the target problems, such as rideshare cab-driver matching or paper-reviewer assignment, may require hundreds or thousands of resources to be concurrently allocated.

1.3 Our Contributions

We formulate our online task assignment problem as a Contextual Restless Multi-armed Bandit (CRMAB) problem. We claim that this model is tractable enough to obtain good algorithms with theoretically provable guarantees and general enough to capture many real-life instances. RMAB and CRMAB problems are known to be difficult to solve optimally, but there is a rich literature developing asymptotically optimal and heuristic policies for these problems [Papadimitriou and Tsitsiklis, 1987, Verloop, 2016, Weber and Weiss, 1990]. Many widely studied RMAB policies are *index* policies [Niño-Mora, 2023]: policies which can characterized by an scalar index function for the value of matching any (task, resource) pair; the index policy assigns those resources to each arriving task with highest index value (independent of the state of other resources).

In this paper, we begin by deriving a new upper-bound LP, OCC-LP, for the expected reward achieved by any assignment policy in the online task assignment problem. This LP is based on occupancy measure flow constraints for our CRMAB model. Using this LP, we propose two index policies for our problem:

Lagrangian index (LAG). Taking the dual of OCC-LP introduces task-wise Lagrange multipliers and decomposes our CRMAB into independent per-resource bandits. We translate these Lagrange multipliers to task-wise subsidies in our CRMAB and derive an *Lagrangian index policy* by solving the relaxed contexual single-armed bandit problem. This yields the index $L(v, r)$ characterizing the value of each (task v, resource r) pair. Upon arrival of a type v task, the LAG policy greedily assigns the k available resources with the highest index $L(v, r)$ at each step.

We show that the time-average reward achieved by LAG converges to the value of OCC-LP in a mean-field limiting regime in Section 4. Our result follows the methods of classic RMAB mean-field optimality results [Verloop, 2016, Weber and Weiss, 1990]. In these results, index policies are shown to be optimal as the number of identical bandit arms, n grows to infinity. However: these classical results break down in our *contextual* RMAB setting. When a new task (context) arrives each time step, the fluid n-scaling regime introduced by [Weber and Weiss, 1990, Whittle, 1988] does not converge to a deterministic fixed point in the limit of growing scales. Rather, the fluid system may cycle its state space dependent on the context history. We propose a new (m, n) *scaling regime* (Definition 4.1), in which we independently grow both the arrivals per time-step, m, and the fan-out, n, of identical resources (arms) seen by each arrival (context). We show that the time-average reward achieved by LAG converges to the value of OCC-LP in the limit of the (m, n)-scaled system:

THEOREM 1.1. *In the (m, n)-scaled system,*

$$\frac{\text{Reward}_{m,n}(\textit{LAG})}{mn} \;\to\; \text{Reward}^*(\textit{OCC-LP}) \qquad \textit{as } m, n \to \infty.$$

Whittle index (WHI). We also show that our underlying RMAB problem is indexable and derive a closed-form Whittle index policy. Our Whittle index function has an easily-computable closed-form in the model parameters:

$$W(v, r) = s_{vr} - d \sum_{v'} p_{v'} \max(s_{v'r} - s_{vr}, 0).$$

While the Whittle index policy does not have asymptotic optimality guarantees, it empirically matches the performance of LAG in real application traces and most problem parameters. Further,

it scales to practical problem regimes in journal peer review and ride-sharing applications, where it is necessary to efficiently allocate hundreds or thousands of resources concurrently.

We evaluate both policies through a data-driven analysis on openly available datasets. We show that our policies perform significantly better in peer review and ridesharing applications than previously proposed algorithms for the online task assignment problem. We find that LAG and WHI obtain the highest reward among all candidate algorithms in nearly all instances. They consistently obtains 94% to 99% of the OCC-LP upper bound.

2 Related Work

Our work sits at the intersection of three bodies of literature: online matching with reusable resources, restless multi-armed bandits, and our motivating problem of paper-reviewer assignment. We survey each in turn, situating our contributions relative to the closest prior work in each community.

2.1 Online Matching and Assignment

The study of online bipartite matching originates with the seminal work of Karp, Vazirani, and Vazirani [Karp et al., 1990], which established competitive ratio as the dominant analytical framework and remains the foundation of a large literature (see [Huang et al., 2024] for a survey). Subsequent work extended this to stochastic arrivals with known i.i.d. distributions [Feldman et al., 2009, Manshadi et al., 2012], and to edge-weighted matching [Fahrbach et al., 2022], where Fahrbach et al. broke the long-standing $\frac{1}{2}$ barrier in the adversarial setting. These works do not allow resource reuse.

A parallel line of work introduces *reusable* resources, where assigned resources become temporarily unavailable and later return to the pool. Dickerson et al. [Dickerson et al., 2021] and Sumita et al. [Sumita et al., 2022] study this model (the OM-RR-KIID setting) for ridesharing applications, deriving LP-based online algorithms with competitive ratio guarantees; their setting focuses on one-to-one matching. Shanks, Yu, and Jacobson [Shanks et al., 2023] study a stochastic variant with reusable resources and geometric arrivals, but restrict similarity scores to a rank-1 product form $w_{tr} = v_t \cdot w_r$; they obtain a $\frac{1}{2}$-competitive algorithm. Delong et al. [Delong et al., 2024] study vertex-weighted reusable matching under adversarial arrivals, achieving a competitive ratio of ≈ 0.589. Gong et al. [Gong et al., 2022] study online assortment optimization with reusable resources in a revenue management framing—a different objective from ours.

This literature has produced strong worst-case guarantees, and also provides LP-based algorithms which inspire our approach. Our OCC-LP is related to, but strictly tighter than, the KIID-LP of [Dickerson et al., 2021] (Section 3). Further, our work departs from this literature in two ways. First, competitive ratio bounds worst-case performance relative to an offline optimum over adversarially chosen instances; in our stochastic setting with known arrival distributions, asymptotic optimality in a fluid scaling regime is the natural performance criterion, and the LP optimum a more informative benchmark.

First, competitive ratio bounds worst-case performance against a benchmark; in our stochastic setting with known arrival distributions, our fluid LP and mean-field optimal index policy yield policies which achieve higher expected rewards on application traces. Second, existing LP-based algorithms require solving a linear program or running simulation at each timestep. As noted in [Sumita et al., 2022], this is computationally feasible only at small scales (tens to hundreds of resources), whereas journal review and rideshare matching require allocating hundreds to thousands of resources concurrently. We include the ALG-SC-LP heuristic of [Dickerson et al., 2021] as a comparison point in our experiments (Section 5).

2.2 Restless Multi-Armed Bandits

A restless multi-armed bandit (RMAB) [Whittle, 1988] consists of n arms, each an independent Markov chain with state space $\mathcal{S}$. At each step, the decision-maker selects exactly nk arms to *activate*; each arm receives a reward and transitions according to its active or passive dynamics, and the objective is to maximize long-run average reward subject to the per-step budget constraint. Optimal control of this problem is PSPACE-hard [Papadimitriou and Tsitsiklis, 1987], which has motivated a rich literature on efficient heuristics with theoretical guarantees [Niño-Mora, 2023].

Whittle [Whittle, 1988] proposed relaxing the per-step constraint to a time-average constraint via Lagrangian relaxation, and defined the *Whittle index* of a state as the scalar subsidy at which active and passive actions are equally desirable; the resulting policy activates the nk arms with the highest current indices. Weber and Weiss [Weber and Weiss, 1990] proved this policy asymptotically optimal as $n \to \infty$ in a fluid limit, under a global attractor assumption. Verloop [Verloop, 2016] extended these results to heterogeneous arms and multiple arm classes via an LP relaxation approach; our proof in Section 4 follows Verloop's approach directly. We also note Avrachenkov, Borkar, and Shah [Avrachenkov et al., 2026] who also study a *Lagrangian index policy* (LIP) for average-reward RMAB, and prove it asymptotically optimal for RMABs with homogeneous arms.

A *contextual* RMAB (CRMAB) extends this framework: at each step a context is observed that modulates arm rewards and, possibly, transitions. Chen and Hou [Chen and Hou, 2024] introduce a CRMAB model in which the context follows a Markov process, arm transitions depend on the context, and the per-step budget is context-dependent. They also derive a Lagrangian index policy via dual decomposition; though they only demonstrate asymptotic optimality numerically. Our work proves asymptotic optimality of the Lagrangian index policy in a new (m, n)-scaling regime, in which both the number of parallel arrival streams m and the resource fan-out n grow (Section 4). This regime is necessary: in the standard $n \to \infty$ scaling with one arrival per step (considered by [Chen and Hou, 2024]), the per-step context randomness prevents the fluid system from converging to a deterministic fixed point, and the Lagrangian reward may not reach the fluid LP bound.

2.3 Paper Reviewer Assignment

Algorithmic approaches to reviewer assignment have focused primarily on the *conference* setting, where all papers arrive simultaneously and assignment is a one-shot batch optimization. The Toronto Paper Matching System (TPMS) [Charlin and Zemel, 2013] pioneered the use of text-similarity affinity scores between papers and reviewer publications, combined with an integer program to maximize total affinity subject to load constraints; this framework for NLP-based affinity scoring has been widely adopted and forms the basis of our similarity scores in Section 5. PeerReview4All [Stelmakh et al., 2021] extends the batch framework by incorporating fairness constraints to ensure a minimum quality of review for every paper. These batch methods are well-suited to conferences with fixed submission deadlines and have been deployed successfully at scale.

Journal review presents a fundamentally different challenge: papers arrive sequentially and must be assigned within a fixed window of their submission. Retrieval-based approaches treat each paper as a query and retrieve suitable reviewers independently, without modeling future arrivals or reviewer availability [Fang and Zhai, 2007, Zhao and Zhang, 2022]. As noted in [Zhao and Zhang, 2022], this leads to overloading of popular reviewers and degraded assignment quality for papers arriving later. In practice, most journal systems default to greedy assignment, assigning each paper to the currently available reviewers with the highest affinity [Biswas and Hasan, 2007, Fang and Zhai, 2007]—a baseline we compare against throughout Section 5. To our knowledge, this work is the first to model journal reviewer assignment as an online assignment problem with

reusable resources, providing a policy with both theoretical guarantees and demonstrated empirical improvement over the greedy baseline.

3 Proposed Policies

We model this problem as a contextual restless multi-armed bandit (CRMAB), where each resource is a bandit arm and the arriving task type is the context. The optimal policy can be computed via dynamic programming, but the state space is $(d+1)^R$, making exact optimization intractable. We develop an occupancy-measure LP upper bound (Section 3.1) and derive two index policies: the Lagrangian index policy Lag (Section 3.2), which uses per-task-type dual prices from the OCC-LP, and a Whittle index policy (Section 3.3) that admits a closed-form expression and scales to large instances. Asymptotic optimality of Lag is proved in Section 4.

Arm Every resource $r \in [R]$ corresponds to a bandit arm. Arm r has state $x_r(t) \in \{0, 1, \ldots, d\}$, where $x_r(t) = i$ means resource r will next become free at $t + i$.

Context At every time step t, a new task v_t arrives, having type v with probability p_v drawn i.i.d. in time.

Action $a_r(t) \in \{0, 1\}$ denotes whether resource r is assigned at time t. At every time step, we must assign k resources to the arriving task, namely we must pull exactly k arms.

$$\forall t, \qquad \sum_{r \in [R]} a_r(t) = k.$$

Transitions For each $r \in [R]$:

$$x_r(t+1) = \begin{cases} d & \text{if } r \in \mathcal{A}_t, \\ \max(x_r(t) - 1,\ 0) & \text{otherwise.} \end{cases}$$

Reward A policy π selects, at each time t, an assignment set $\mathcal{A}_t \subseteq \{r \in [R] : x_r(t) = 0\}$ with $|\mathcal{A}_t| = k$. The assignment is immediate and irrevocable. Assigning $\mathcal{A}_t$ to paper v_t yields reward $\sum_{r \in \mathcal{A}_t} s_{v_t r}$. The long-run average reward of policy π is

$$\text{Reward}(\pi) \ := \ \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_\pi \left[\sum_{r \in \mathcal{A}_t} s_{v_t r} \right].$$

Since the bandits are Markovian, it suffices to consider policies which map system state (the state of all arms, and the current context) to feasible action vectors, namely $\pi : \{0, 1, \ldots, d\}^R \times [V] \to \{0, 1\}^R$.

3.1 OCC-LP, the Occupancy Measure LP upper bound

We examine the Occupancy Measure LP of the CRMAB introduced in Section 3, whose optimal value upper-bounds Reward(π) for any policy π (Proposition 3.1). The decision variables are $u_{vr} \geq 0$ for $v \in [V]$, $r \in [R]$, representing the long-run rate of assignments of resource r to type-v tasks, and

Table 1. Summary of notation used in Section 3.

Symbol	Description
Problem parameters	
R	Number of resource types
V	Number of task types
d	Service duration (steps a resource is unavailable after assignment)
k	Number of resources assigned per arriving task
$s_{vr} \in \mathbb{R}$	Match quality of resource r for task type v
p_v	Arrival probability of task type v, i.i.d. over time
CRMAB model	
$x_r(t) \in \{0, \dots, d\}$	State of resource r at time t; $x_r(t) = 0$ means available
$\mathcal{A}_t \subseteq [R]$	Set of k resources assigned at time t
$\mathrm{Reward}(\pi)$	Long-run average reward of policy π
OCC-LP variables	
$u_{vr} \geq 0$	Long-run rate of assigning resource r to type-v tasks
$\phi_r \geq 0$	Steady-state free fraction of resource r
Lagrangian index	
ℓ_v^*	Dual variable for task-v assignment-rate constraint (2)
q_r^*	Dual variable (optimal Arm-r LP value) for constraint (3)
$L(v, r) = s_{vr} - \ell_v^* - d\, q_r^*$	Lagrangian index of resource r for task type v
Whittle index	
$\ell \in \mathbb{R}$	Uniform scalar penalty (replaces per-type ℓ_v^*)
$q_r^*(\ell)$	Time-average value of Arm-r bandit under penalty ℓ
$W(v, r) = s_{vr} - d\sum_{v'} p_{v'}(s_{v'r} - s_{vr})^+$	Whittle index of resource r for task type v

auxiliary variables $\phi_r \geq 0$ for $r \in [R]$, representing the steady-state free fraction of resource r.

$$\max_{u,\phi} \quad \sum_{v\in[V],\, r\in[R]} s_{vr}\, u_{vr} \tag{OCC-LP}$$

$$\text{s.t.} \quad \sum_{r\in[R]} u_{vr} = k\, p_v \qquad \forall\, v \in [V] \tag{2}$$

$$\phi_r = 1 - d \sum_{v\in[V]} u_{vr} \qquad \forall\, r \in [R] \tag{3}$$

$$0 \leq u_{vr} \leq p_v\, \phi_r \qquad \forall\, v \in [V],\ r \in [R] \tag{4}$$

Constraint (2) requires that type-v tasks receive k resource assignments per unit time. Constraint (3) defines ϕ_r via Little's Law: each assignment to resource r occupies it for d steps at total rate $\sum_{v\in[V]} u_{vr}$, so the busy fraction is $d \sum_{v\in[V]} u_{vr}$. Constraint (4) ensures the assignment rate for pair (v, r) does not exceed the rate at which resource r is free and a type-v task arrives.

Proposition 3.1 (LP upper bound). *For any policy π,* $\mathrm{Reward}(\pi) \leq \mathrm{Reward}^*(\textit{OCC-LP})$.

Proof. Fix a policy π. Let $\phi_r^\pi := \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbf{1}[x_r^\pi(t) = 0]$ be the long-run free fraction of resource r and let

$$u_{vr}^\pi \;:=\; \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_\pi[\mathbf{1}[v_t = v,\ r \in \mathcal{A}_t]]\,, \qquad v \in [V],\ r \in [R].$$

The time average reward obtained by π is

$$\text{Reward}(\pi) = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{r\in[R]} \sum_{v\in[V]} \mathbf{1}[v_t = v,\ r \in \mathcal{A}_t] \cdot s_{vr} = \sum_{r\in[R]} \sum_{v\in[V]} u^{\pi}_{vr} s_{vr}.$$

We show (u^{π}, ϕ^{π}) is feasible for (OCC-LP).

Feasibility of (2): At each time t, exactly k resources are assigned to task v_t, so $\sum_{r\in[R]} \mathbf{1}[r \in \mathcal{A}_t] = k$. Thus,

$$\begin{aligned}
\sum_{r\in[R]} u^{\pi}_{vr} &= \sum_{r\in[R]} \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbf{1}[v_t = v,\ r \in \mathcal{A}_t] \\
&= \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{r\in[R]} \mathbf{1}[v_t = v,\ r \in \mathcal{A}_t] \\
&= \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} k \cdot \mathbf{1}[v_t = v] \\
&= k \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbf{1}[v_t = v] = k p_v.
\end{aligned}$$

Feasibility of (3): Each assignment to resource r occupies it for d consecutive timesteps. The total assignment rate to resource r is $\sum_{v\in[V]} u^{\pi}_{vr}$. In particular, suppose $u_{vr}(t) = \mathbf{1}[v_t = v, r \in \mathcal{A}_t]$,

$$\begin{aligned}
\phi^{\pi}_r &= \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbf{1}[x^{\pi}_r(t) = 0] \\
&= \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} (1 - \sum_{t'=\min(1,t-d)}^{t-1} \sum_{v\in[V]} u^{\pi}_{vr}(t')) \\
&= 1 - \sum_{v\in[V]} \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{t'=\min(1,t-d)}^{t-1} u^{\pi}_{vr}(t') \\
&= 1 - \sum_{v\in[V]} \lim_{T\to\infty} \frac{1}{T} \sum_{t'=1}^{T-1} \min(d, T-t') u^{\pi}_{vr}(t') \\
&= 1 - d \sum_{v\in[V]} u^{\pi}_{vr}.
\end{aligned}$$

Feasibility of (4): Resource r can be assigned to a type-v task only when $v_t = v$ and $x_r(t) = 0$.

$$u^{\pi}_{vr} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[\mathbf{1}[v_t = v,\ r \in \mathcal{A}_t]] \le \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[\mathbf{1}[v_t = v,\ x_r(t) = 0]]$$

Let $\mathcal{F}_t = \{(v_s, \mathcal{A}_s)\}_{s\le t}$ be the filtration on arrivals and assignments upto t. Let $M_T = \sum_{t=1}^{T} (\mathbf{1}[v_t = v] - p_v) \cdot \mathbf{1}[x_r(t) = 0]$. M_T is a martingale since

$$\mathbb{E}[M_T | \mathcal{F}_{t-1}] = M_{T-1} + \mathbb{E}[(\mathbf{1}[v_t = v] - p_v) \cdot \mathbf{1}[x_r(t) = 0] | \mathcal{F}_{t-1}] = M_{T-1}.$$

Thus $\lim_{T\to\infty} \frac{1}{T} M_T \to 0$. This implies

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[\mathbf{1}[v_t = v,\ x_r(t) = 0]] = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} p_v \mathbf{1}\left[x_r(t) = 0\right] = p_v \phi_r^{\pi}.$$

Thus we have $0 \le u_{vr}^{\pi} \le p_v \phi_r^{\pi}$.

Therefore (u^{π}, ϕ^{π}) is feasible for (OCC-LP). The long-run average reward

$$\text{Reward}(\pi) = \sum_{v\in[V],\, r\in[R]} s_{vr}\, u_{vr}^{\pi} \le \text{Reward}^*(\text{OCC-LP}).$$

□

Note that this upper bound is similar to, but tighter than, the KIID variant of [Dickerson et al., 2021, LP (1)].

3.2 Lag, the Lagrangian Index Policy

We derive a Lagrangian index policy via dual decomposition of the OCC-LP. We show in Section 5 that it achieves 94–99% of the OCC-LP upper bound across all evaluated instances, and prove asymptotic optimality in a mean field scaling regime discussed in Section 4.

Lagrangian decomposition. Form the Lagrangian by relaxing constraints (2) with multipliers $\ell_v \in \mathbb{R}$, one per task type $v \in [V]$:

$$\begin{aligned}\mathcal{L}(u, \ell) &= \sum_{v\in[V],\, r\in[R]} s_{vr}\, u_{vr} + \sum_{v\in[V]} \ell_v \left(k\, p_v - \sum_{r\in[R]} u_{vr} \right) \\ &= \sum_{r\in[R]} \sum_{v\in[V]} (s_{vr} - \ell_v)\, u_{vr} + \sum_{v\in[V]} \ell_v\, k\, p_v,\end{aligned}$$

subject to (3) and (4). For fixed ℓ, the maximization over u decouples across resources $r \in [R]$. For each $r \in [R]$, the Arm-r LP is:

$$\begin{aligned}\max_{(u_{vr})_{v\in[V]}} \quad & \sum_{v\in[V]} (s_{vr} - \ell_v)\, u_{vr} && \text{(Arm-}r\text{ LP)} \\ \text{s.t.} \quad & 0 \le u_{vr} \le p_v \phi_r \qquad \forall\, v \in [V] \\ & \phi_r = 1 - d \sum_{v\in[V]} u_{vr}.\end{aligned}$$

Let $q_r(\ell)$ denote the optimal value of the Arm-r LP.

Dual LP. The Lagrangian dual problem is:

$$\min_{\ell\in\mathbb{R}^V} \sum_{r\in[R]} q_r(\ell) + \sum_{v\in[V]} \ell_v\, k\, p_v.$$

By strong duality, the optimal dual value equals OPT(OCC-LP). Let $\ell^* = (\ell_v^*)_{v\in[V]}$ be an optimal dual solution, (u^*, ϕ^*) the corresponding primal solution, and $q_r^* := q_r(\ell^*)$.

Definition 3.2 (Lag Policy.). For each $v \in [V]$ and $r \in [R]$, define the index:

$$L(v, r) \ :=\ s_{vr} - \ell_v^* - d\, q_r^*. \tag{5}$$

The Lag policy is: when a task of type $v \in [V]$ arrives, assign it to the k available resources with the largest values of $L(v, r)$, breaking ties arbitrarily.

The Lagrangian index is derived based on the following principle: We relax the constraint that we must assign exactly k resources to each arrival. Instead, we we make every (v, r) assignment where $L(v, r) > 0$ whenever it is valid. This decoupled policy (Lag-Relax) achieves time-average reward equal to the value of OCC-LP. The following proposition makes this precise:

Proposition 3.3. *The solution to OCC-LP, (u^*, ϕ^*) satisfies for every $v \in [V], r \in [R]$,*

$$\begin{aligned} L(v,r) > 0 &\implies u^*_{vr} = p_v \phi^*_r, \\ L(v,r) = 0 &\implies u^*_{vr} \in [0, p_v \phi^*_r], \\ L(v,r) < 0 &\implies u^*_{vr} = 0. \end{aligned}$$

In the real model, the Lag index will be used to assign exactly k assignments to each arrival, so this decoupled optimality result does not translate to our original setting. In the following section, we will establish an asymptotic scaling regime in which this distinction between the original and relaxed problems vanishes.

However, we first establish the relaxed optimality criteria to motivate and derive our index formulation. Below, to prove Proposition 3.3, we translate Arm-r LP to a bandit problem. Lemma 3.5 shows the time-average bandit value equals the value of Arm-r LP, q^*_r. Lemma 3.6 shows the optimal bandit policy accepts in state $(0, v)$ if and only if $L(v, r) \geq 0$. Proposition 3.3 follows by combining the two lemmas.

Definition 3.4 (Arm-r bandit). Given a vector or service penalties $\ell \in \mathbb{R}^V$, we define the Arm-r bandit as follows:

State describes the time to next availability, $x \in \{0, 1, \ldots, d\}$ and $v \in [V]$, the current context/task-type.

Action $a \in \{0, 1\}$, for whether to accept or reject in a current state. We can only accept when the resource is free, i.e. $x = 0$. If we accept in state $(0, v)$, we get reward $s_{vr} - \ell_v$. Further, the resource becomes busy for the next d steps, i.e. it transitions to state $x' = d$. If we reject, we get reward 0, transition to state with $x' = 0$ and v' drawn as an i.i.d. task-type.

Objective Given a policy $\pi : \{0, 1, \ldots, d\} \times [V] \to \{0, 1\}$, let $V^{r,\gamma}_\pi(x, v)$ denote the γ-discounted value starting from state (x, v) and let $\bar{V}^{r,\gamma}_\pi(x) = \mathbb{E}_v\left[V^{r,\gamma}_\pi(x, v)\right] = \sum_v p_v V^{r,\gamma}_\pi(x, v)$. The objective is to maximize over π

$$V^{r,\gamma}_\pi(x, v) = (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t (s_{vr} - \ell_v) \mathbf{1}[x_\pi(t) = 0, a_\pi(t) = 1]$$

starting from $x(0) = x, v(0) = v$. Note that as $\gamma \to 1$, the value of state (x, v) converges to the time average value. We call this the time-average Arm-r bandit.

The bandit reward $s_{vr} - \ell_v$ decomposes into the raw match score s_{vr} minus a task-type-wise penalty ℓ_v. The penalty ℓ^*_v is the shadow price of serving type-v tasks: it reflects the opportunity cost of occupying a resource for a type-v arrival, as determined by the global per-type demand constraints (2).

Lemma 3.5 (Arm-r Bandit-LP equivalence). *For every $x \in \{0, 1, \ldots, d\}, v \in [V]$, the value of the time-average Arm-r bandit equals the value of the Arm-r LP.*

$$\lim_{\gamma \to 1} V^{r,\gamma}_*(x, v) = q^*_r.$$

Proof. The upper bound ($\lim_{\gamma \to 1} V^{r,\gamma}_*(x, v) \leq q^*_r$.) follows similar to Proposition 3.1. For every arm-r policy π, the time-average occupancy measure of $(x = 0, v, a = 1)$ under π represents the

variable u^π_{vr}. We have that for every v, $0 \le u^\pi_{vr} \le p_v \phi^\pi_r$. In the Arm-$r$ bandit, the time average reward is $\sum_{vr}(s_{vr} - \ell_v)u^\pi_{vr}$. Thus every bandit policy corresponds to feasible point of the Arm-r LP.

To show equality, we begin with the LP solution (u^*, ϕ^*) and construct a bandit policy π achieving this value as follows: when a type v task arrives and the r resource is free, it accepts with probability $\frac{u^*_{vr}}{p_v \phi^*_r}$. Then we have occupancy measure

$$\begin{aligned} u^\pi_{vr} &= \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbf{1}[x^\pi_r(t) = 0, v(t) = v, a^\pi_r(t) = 1] \\ &= \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \Pr[x^\pi_r(t) = 0, v(t) = v] \cdot \frac{u^*_{vr}}{p_v \phi^*_r} \\ &= \frac{u^*_{vr}}{\phi^*_r} \cdot \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \Pr[x^\pi_r(t) = 0] \\ &= \frac{u^*_{vr}}{\phi^*_r} \cdot \phi^\pi_r. \end{aligned}$$

Following a Little's Law argument as in Proposition 3.1, we have $\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \Pr[x^\pi_r(t) = 0] = 1 - d \sum_v u^\pi_{vr} = \phi^\pi_r$. Then,

$$\phi^\pi_r = 1 - d \sum_v u^\pi_{vr} = 1 - \frac{\phi^\pi_r}{\phi^*_r} \cdot d \sum_v u^*_{vr} = 1 - \frac{\phi^\pi_r}{\phi^*_r} \cdot (1 - \phi^*_r) = \phi^\pi_r + 1 - \frac{\phi^\pi_r}{\phi^*_r}.$$

Thus we have $\phi^\pi_r = \phi^*_r$ and $u^\pi_{vr} = u^*_{vr}$, so the time-average reward of π is $\sum_v (s_{vr} - \ell_v) u^*_{vr} = q^*_r$. □

Lemma 3.6. *The optimal policy for the time-average Arm-r bandit accepts in state $(0, v)$ iff*

$$\begin{aligned} s_{vr} - \ell_v > d q^*_r &\implies \pi^*(0, v) = 1. \\ s_{vr} - \ell_v = d q^*_r &\implies \pi^*(0, v) \in \{0, 1\}. \\ s_{vr} - \ell_v < d q^*_r &\implies \pi^*(0, v) = 0. \end{aligned}$$

Proof. The Bellman equation for the γ-discounted Arm-r bandit accepts in state $(0, v)$ can be stated as

$$V^{r,\gamma}_*((0, v)) = \max((1 - \gamma)(s_{vr} - \ell_v) + \gamma^{d+1} \bar{V}^{r,\gamma}_*(0), \gamma \bar{V}^{r,\gamma}_*(0))$$

Here, the optimal policy accepts in state $(0, v)$, i.e. $\pi^*(0, v) = 1$ iff

$$s_{vr} - \ell_v \ge \frac{\gamma(1 - \gamma^d)}{1 - \gamma} \bar{V}^{r,\gamma}_*(0)$$

with indifference at equality. Taking the limit $\gamma \to 1$, $\pi^*(0, v) = 1$ iff $s_{vr} - \ell_v \ge d \lim_{\gamma\to 1} \bar{V}^{r,\gamma}_*(0) = d q^*_r$ by Lemma 3.5. □

Lemma 3.6 applied to the dual optimal penalties ℓ^* tells us that the optimal Arm-r bandit policy with penalties ℓ^* accepts type v tasks iff $L(v, r) > 0$ (and is indifferent to accepting tasks with $L(v, r) = 0$). Lemma 3.5 tells us it achieves time-average reward equal to the value of Arm-r LP with penalties ℓ^*. This value coincides with value of OCC-LP. Further, the occupancy measure of this policy is the solution to OCC-LP, as shown in Lemma 3.5. Thus we establish Proposition 3.3.

3.3 Whi, the Whittle Index Policy

The Lag index requires solving the OCC-LP (for ℓ_v^*) and per-arm LPs (for q_r^*), which is feasible only for small problem sizes. In practice, reviewer assignment involves hundreds of reviewers, and ridesharing platforms dispatch thousands of drivers. LP-based approaches cannot scale to these regimes. Sumita et al. [2022] note this gap in their future work, calling for scalable policies that avoid LP computation. Most automated online reviewer assignment systems currently default to greedy assignment (defined precisely in Section 5.1) for precisely this reason [Fang and Zhai, 2007, Zhao and Zhang, 2022].

To overcome these issues, we apply the Whittle index methodology [Whittle, 1988], replacing the per-type dual prices ℓ_v^* with a uniform scalar penalty ℓ across all task types in the Arm r bandit. Let $q_r^*(\ell)$ denote the time-average value of the Arm-r bandit under this penalty. In the Arm-r bandit, busy states ($x > 0$) are always passive. The relevant states are $(0, v)$, where the resource is available and a type-v task arrives. Let $P_r(\ell)$ denote the set of states in which passivity is optimal under the ℓ-penalized Arm-r bandit problem. The bandit is *indexable* if $P_r(\ell)$ is non-decreasing in ℓ. For an indexable bandit, the *Whittle index* of state $(0, v)$ is the *penalty of indifference*:

$$W(v, r) \;:=\; \inf\big\{\, \ell \in \mathbb{R} : (0, v) \in P_r(\ell) \,\big\},$$

the smallest penalty at which passivity becomes optimal in state $(0, v)$ in the Arm r bandit. The Whittle index of state $(0, v)$ in the Arm-r bandit is the penalty ℓ at which accepting type v and remaining passive are equally valuable.

We prove that the Arm-r bandit with uniform service penalties is indexable (Theorem 3.7). Further, we derive a closed form expression for the Whittle index (Theorem 3.8).

Theorem 3.7 (Indexability). *The Arm-r bandit with uniform scalar penalty ℓ is indexable: the passive set $P_r(\ell)$ is non-decreasing in ℓ.*

Theorem 3.8 (Whittle index formula). *The time-average Whittle index for assigning resource r when it is available and a task of type v arrives is*

$$W(v, r) \;=\; s_{vr} - d \sum_{v' \in [V]} p_{v'} \left(s_{v'r} - s_{vr}\right)^+. \tag{6}$$

Definition 3.9 (Whi policy). The Whi policy is defined as follows. when a task of type $v \in [V]$ arrives, assign it to the k available resources with the largest values of $W(v, r)$, breaking ties arbitrarily.

Below, we present the proofs of Theorems 3.7 and 3.8.

We begin using Lemma 3.6 applied to uniform penalty ℓ. The optimal policy in the Arm-r bandit with uniform service penalty ℓ accepts tasks of type v iff $s_{vr} - \ell > q_r^*(\ell)$. We compute $q_r^*(\ell)$ in terms of ℓ. This yields the following Lemma characterizing the optimal policy for the Arm-r bandit with uniform service penalty ℓ.

Lemma 3.10. *The optimal policy for the Arm-r bandit with uniform service penalty ℓ accepts tasks of type v iff $\tilde{W}(v, r) > \ell$, where*

$$\tilde{W}(v, r) := s_{vr} - d \sum_{v' \in [V]} p_{v'} \left(s_{v'r} - s_{vr}\right)^+.$$

Proof. Let $p = \sum_{v' : s_{v'r} \geq s_{vr}} p_{v'}$ and $\bar{s} = \sum_{v' : s_{v'r} \geq s_{vr}} p_{v'}\, s_{v'r}$. We compute $q_r^*(\ell)$, the time-average value of Arm r bandit as follows: we first evaluate the discounted Arm-r bandit value under the threshold policy $\pi_{s_{vr}}$ that accepts type v' if and only if $s_{v'r} \geq s_{vr}$, then take $\gamma \to 1$ as in Lemma 3.5.

Let $p = \sum_{v': s_{v'r} \geq s_{vr}} p_{v'}$ and $\bar{s} = \sum_{v': s_{v'r} \geq s_{vr}} p_{v'} s_{v'r}$. Under $\pi_{s_{vr}}$ with uniform penalty ℓ and discount factor γ:

$$\bar{V}^{r,\gamma}(0) = \sum_{v': s_{v'r} \geq s_{vr}} p_{v'} \left[(1-\gamma)(s_{v'r} - \ell) + \gamma^{d+1} \bar{V}^{r,\gamma}(0) \right] + \sum_{v': s_{v'r} < s_{vr}} p_{v'} \cdot \gamma \, \bar{V}^{r,\gamma}(0).$$

Solving:

$$\bar{V}^{r,\gamma}(0) = \frac{(1-\gamma)(\bar{s} - p\ell)}{1 - (1-p)\gamma - p\gamma^{d+1}}.$$

As $\gamma \to 1$, using $\gamma(1-\gamma^d) \to d(1-\gamma)$ and $1 - (1-p)\gamma - p\gamma^{d+1} \to (1-\gamma)(1+dp)$:

$$q_r^*(\ell) = \lim_{\gamma \to 1} \bar{V}^{r,\gamma}(0) = \frac{\bar{s} - p\ell}{1 + dp}.$$

Thus the optimal policy for the Arm-r bandit with uniform service penalty ℓ accepts tasks of type v iff

$$\begin{aligned} & (s_{vr} - \ell)(1 + dp) \geq d(\bar{s} - p\ell) \\ \iff \quad & s_{vr} - d(\bar{s} - p s_{vr}) \geq \ell \\ \iff \quad & \tilde{W}(v, r) \geq \ell. \end{aligned}$$

□

Theorems 3.7 and 3.8 follow from Lemma 3.10: We have that set of free states in which the passive action is optimal,

$$P_r(\ell) \cap \{(0, v) : v \in [V]\} = \{(0, v) : \tilde{W}(v, r) \leq \ell\}.$$

This is clearly non-decreasing in ℓ, thus the Arm-r bandit is indexable. Further

$$W(v, r) = \inf\{ \ell \in \mathbb{R} : (0, v) \in P_r(\ell) \} = \inf\{ \ell \in \mathbb{R} : \tilde{W}(v, r) \leq \ell \} = \tilde{W}(v, r).$$

This completes the proof.

4 Asymptotic Optimality of Lag

In this section, we propose a mean-field scaling regime and prove that Lag is asymptotically optimal in the scaling regime.

4.1 The Mean Field Scaling Regime

A rich body of prior work studies mean-field asymptotic optimality of index policies for restless bandit problems [Larrañaga, 2015, Niño-Mora, 2023, Verloop, 2016, Weber and Weiss, 1990]. The classical result of Weber and Weiss [1990] shows that when there are n identical arms, the Whittle index policy is optimal as $n \to \infty$. However, these results do not directly extend to our *contextual* RMAB setting. It is not immediately clear how a CRMAB should be scaled. We begin by considering two natural approaches:

Batch-scaling (n-scaling). Suppose we scale the number of resources by a factor of n, so that there are n copies of each resource type. At each time step, a single arrival v_t is assigned nk resources out of a pool of size nR. This model is depicted in Figure 2b. Chen and Hou [2024] consider a related scaling regime for general CRMABs.

The difficulty is that the arrival process remains stochastic at unit rate. As a result, even as $n \to \infty$, the system does not converge to a deterministic fixed point. Instead, the evolution of the system depends on the realized sequence of contexts, and the fluid limit may exhibit non-trivial dynamics (e.g., limit cycles) driven by the context process.

Buffet-scaling (m-scaling). Alternatively, suppose there are m i.i.d. arrivals at each time step, and m copies of each resource type. Each arrival must be matched to k free resources, thus a total of mk assignments are made at each step. This model is depicted in Figure 2c.

This avoids the limit-cycle issue, since averaging over many arrivals produces a deterministic system. However, this scaling fundamentally changes the problem. At each time step, assignments can be made jointly across all arrivals, allowing resources to effectively choose the most favorable matches among a batch of tasks. This is strictly more powerful than in the original problem, where each resource only observes a single arrival.

Formally, under this scaling, the induced occupancy measures (u^π, ϕ^π) may violate the constraint (4) in OCC-LP. The limiting LP (e.g., the KIID LP [Dickerson et al., 2021]) therefore provides a looser upper bound. In our experiments, heuristics based on this LP can perform worse, likely because the relaxation models the problem less tightly. This is discussed in Section 5.

Thus, neither scaling provides the right generalization of the Weber–Weiss regime: n-scaling preserves the problem structure but does not yield a deterministic limit, while m-scaling yields a deterministic limit but relaxes the problem.

The (m, n)-scaling. We propose a scaling that combines both ideas. We scale the number of arrivals and resources, while ensuring that each resource only observes one arrival per time step.

At each time step, m i.i.d. arrivals occur, and there are mn resources of each type. For each resource type, the mn resources are uniformly partitioned into m groups of size n, and each group is assigned to one arrival. Thus, each arrival observes n resources of each type (for a total of nR), and must select nk of them. Assigned resources become unavailable for the next d steps.

We scale the arrival rate by m, the demand per arrival by n, and the resource pool by mn, so that the system load remains unchanged:

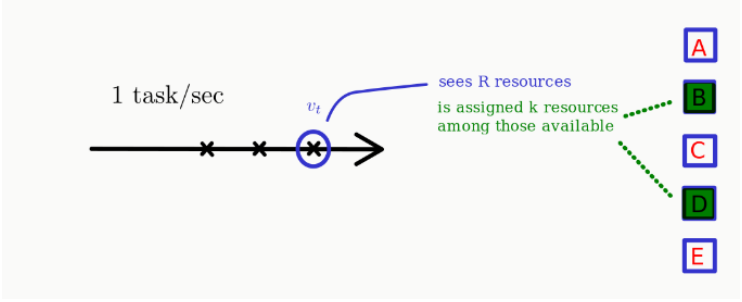
$$\rho = \frac{kd}{R}.$$

Under this scaling, the system admits a mean-field limit as $m, n \to \infty$. The randomness in arrivals and resource states averages out, and each arrival effectively sees a deterministic fraction of available resources of each type. At the same time, the per-resource interaction structure of the original problem is preserved.

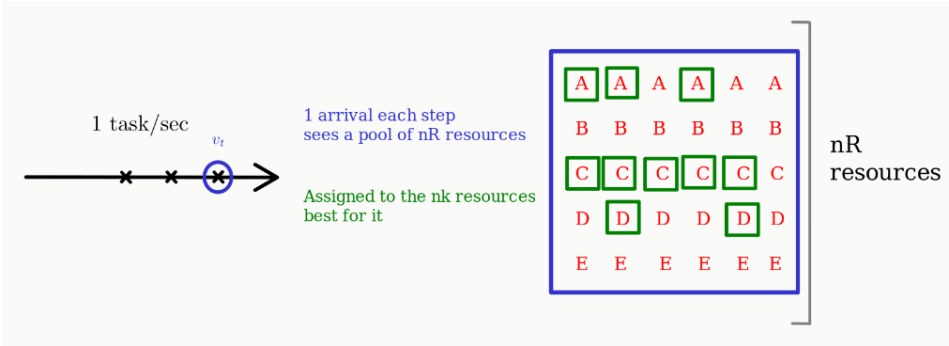
The (m, n)-scaling admits a natural operational interpretation as a large-scale, distributed matching system with limited visibility, depicted in Figure 3.

(1) First, the m arrivals at each time step can be viewed as m independent *dispatchers* (or frontends), each receiving one task. Rather than a single centralized decision-maker with access to the entire resource pool, decisions are made locally and in parallel.
(2) Second, the random partitioning of the mn resources into m groups of size n models *limited access to resources.* Each dispatcher only observes a subset of size n of each resource type, rather than the full system of size mn. This reflects practical constraints in large systems, where querying or coordinating across all resources is infeasible due to latency or computational overhead. The partitioning can be interpreted as randomized load balancing or hashing of resources to dispatchers, refreshed at each time step.
(3) Third, each resource is queried by exactly one dispatcher per time step. This enforces that a resource is matched based only on a *single* arriving context, as in the original problem. In particular, resources cannot compare across multiple simultaneous arrivals, avoiding the "buffet" effect of m-scaling where resources effectively choose their most preferred task from a batch.

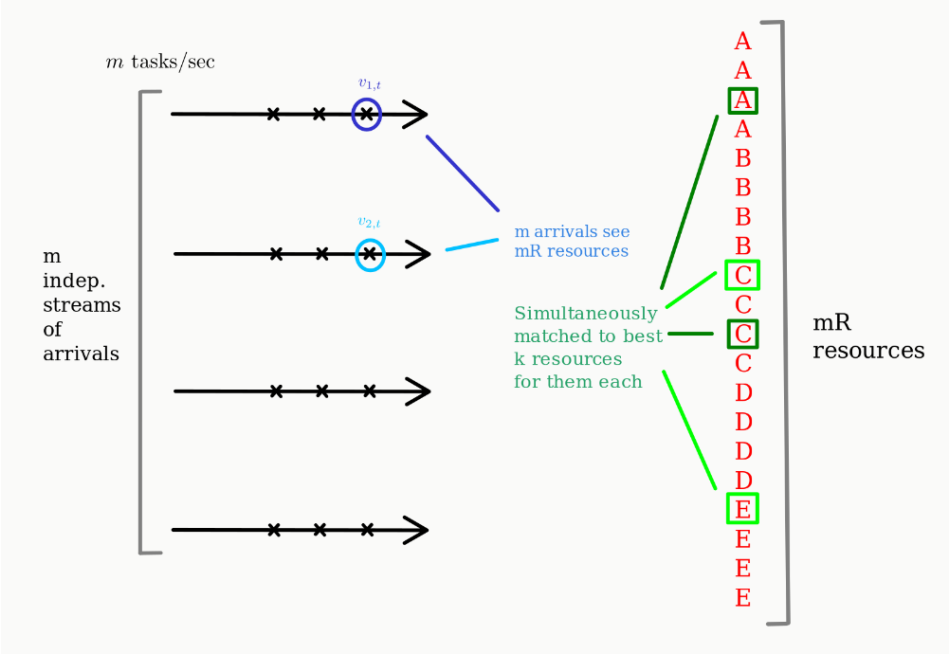
We define (m, n)-scaling formally below.



(a) Original system



(b) *n*-scaled system



(c) *m*-scaled system

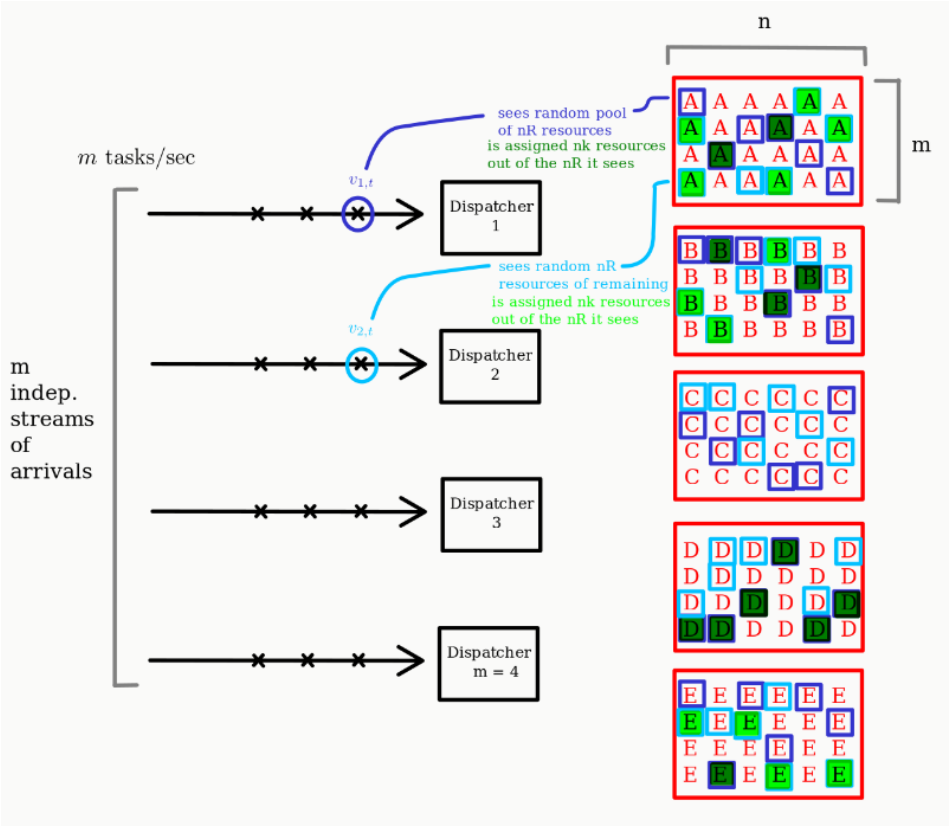


Fig. 3. (m, n)-scaled system (m=4, n=6, R=5, k=2): m independent streams share mn=24 copies of each resource type (A–E), split into m groups of n each per timestep. Blue cell = Dispatcher$_1$'s partition (nR=30 total); green cell = assigned this step (nk=12 total)

Definition 4.1 ((m, n)-scaled system). Fix $m, n \in \mathbb{N}$. The (m, n)-*scaled system* is defined as follows.

Resources. For each type $r \in [R]$, there are mn resources of type r.
Arrivals. For each timestep t, m i.i.d. tasks arrive
Distribution. The mn resources of each type are split into m uniform partitions, such that every resource sees a uniformly chosen arrival and each arrival sees n uniformly chosen resources of each type.
Assignment. Every arrival is assigned nk resources out of the nR distributed to it, or as many as are available.

For any policy π, let $\text{Reward}_{m,n}(\pi)$ denote the long-run average total reward per timestep under π in the (m, n)-scaled system. The Lag index assigns each arriving task $v_{j,t}$ for $j \in [m]$ to the nk available resources with the largest index $L(v_{j,t}, r)$ out of the nR resources distributed to it.

In the remainder of the section, we show that the time-average reward achieved by Lag converges to the value of OCC-LP in the limit of the (m, n)-scaled system. Since OCC-LP was an upper bound on the expected reward of any policy (Proposition 3.1), this means Lag is asymptotically optimal. We prove the following Theorem:

Theorem 4.2 (Asymptotic optimality of Lag). *Given Assumption 4.10,*

$$\frac{\text{Reward}_{m,n}(\textit{LAG})}{mn} \;\to\; \text{Reward}^*(\textit{OCC-LP}) \qquad \textit{as } m,n \to \infty.$$

The dynamics of the (m,n)-scaled system can be described using the following state representation: $(x_{r,i}(t))_{i\in[d],r\in[R]}$ where $x_{r,i}(t)$ denotes the fraction of type-r resources in state $i \in \{0,1,\dots,d\}$. Thus for each $r \in [R]$, $t \in [T]$, we have $\sum_{i=0}^{d} x_{r,i}(t) = 1$. For the (m,n)-scaled system, we must have $x_{r,i}(t) \in \{0, \frac{1}{mn}, \dots, 1\}$ for each r, i, t, since $x_{r,i}(t)$ represents the fraction of type r resources in state i, out of a total mn. Thus, the (m,n)-scaled system under Lag is a finite Discrete-Time Markov Chain (DTMC) with state space

$$\mathcal{X}^{m,n} = \left\{ x \in [0,1]^{R\times(d+1)} : \forall r,i, x_{r,i} \in \{0, \frac{1}{mn}, \dots, 1\}, \forall r, \sum_{i=0}^{d} x_{r,i} = 1 \right\}.$$

Every finite DTMC has a stationary distribution, we will denote the stationary distribution of the (m,n)-scaled system under Lag as $\mu^{m,n} \in \mathbb{P}(\mathcal{X}^{m,n})$, where $\mathbb{P}(S)$ is used to denote the set of probability distributions over finite set S.

4.2 Lemmas concerning the waterfill function

Definition 4.3. (Waterfilling) Given a budget b and vector $c \in [0,1]^R$ of capacity fractions: c_r for each $r \in [R]$, allocation $u \in \mathbb{R}^R$ is said to be a waterfill of (b, c, π) iff

(1) No allocation exceeds capacity: $0 \le u_r \le c_r$ for every r
(2) Budget is fulfilled: $\sum_r u_r = b$; or all capacities are used, $u = c$.
(3) For all r, r': $\pi(r) < \pi(r')$ and $u_r < c_r \implies u_{r'} = 0$.

We denote this as $u = \text{waterfill}(b, c, \pi)$ and $u_r = \text{waterfill}_r(b, c, \pi)$.

Plainly, $\text{waterfill}(b, c, \pi)$ fills capacities $c_{\pi(1)}, c_{\pi(2)}, \dots$ until the budget is filled or the capacities are exhausted. Thus suppose π is the priority order $1 > 2 > \cdots > R$, there is $r \in [R]$ such that $\text{waterfill}(b, c, \pi) = (c_1, \dots, c_{r-1}, b - \sum_{r'<r} c_{r'}, 0, \dots, 0)$. We show that waterfill is Lipschitz continuous in the capacity vector.

Lemma 4.4. *For any fixed b, π, the map $f : [0,1]^R \to [0,1]^R$ defined as $x \mapsto \text{waterfill}(b, x, \pi)$ is 2-Lipschitz with respect to the ℓ_1-norm.*

Proof. Without loss of generality, let the priority order π be $1 > 2 > \cdots > R$. Let $x \in [0,1]^R$. We begin by perturbing a single index of x. Let $x' = x + \delta \cdot e_j$, where e_j is the basis vector with 1 at index j and zero elsewhere and $\delta > 0$. That is, suppose we increase only the capacity of the j^{th} resource by δ. Then the allocation to all higher priority resources is unchanged. The allocation to resource j, $f_j(x)$ may increase, by at most δ. The allocation to all lower priority resources may only decrease. Further, the total sum of allocations must stay equal to b, or may increase if the capacities were originally exhausted. Thus the total decrease in allocation to all lower priority resources is at most the increase in allocation to resource j. Thus

$$\begin{aligned}\|f(x') - f(x)\|_1 &= \sum_{i=1}^{R} |f_i(x') - f_i(x)| = (f_j(x') - f_j(x)) - \sum_{i>j} (f_i(x') - f_i(x)) \\ &\le 2(f_j(x') - f_j(x)) \le 2\delta.\end{aligned}$$

A similar argument follows if $\delta < 0$: the allocation to resource j may decrease by at most $|\delta|$, and correspondingly, the total allocation to all higher priority resources increases by at most $|\delta|$.

Now suppose we have $x' - x = \delta \in [-1, 1]^R$. We perturb one index at a time:

$$\|f(x') - f(x)\|_1 = \|f(x + \sum_{j=1}^{R} \delta_j e_j) - f(x)\|_1$$
$$\leq \sum_{j=1}^{R} \|f(x + \sum_{i=1}^{j} \delta_i e_i) - f(x + \sum_{i=1}^{j-1} \delta_i e_i)\|_1 \leq \sum_{j=1}^{R} 2|\delta_j| = 2\|x' - x\|_1.$$

□

Lemma 4.5. *There exists a* (OCC-LP) *optimum* (u^*, ϕ^*) *such that for each* $v \in [V]$,

$$u^*_{v,r} = p_v \cdot \text{waterfill}_r(k, \phi^*, L(v, \cdot)).$$

That is, $(u^*_{vr})_r$ *is a waterfill of budget* kp_v*, capacities* $(p_v\phi^*_r)_r$*, and ordering* $r \mapsto L(v, r)$.

Proof. By Lemma 3.6, the optimal arm-r policy π^* accepts type v when free iff $L(v, r) \geq 0$. By Lemma 3.5, the occupancy measure (u^*, ϕ^*) of π^* achieves the arm-r LP optimum, hence satisfies:

$$L(v, r) > 0 \implies u^*_{vr} = p_v\phi^*_r,$$
$$L(v, r) = 0 \implies u^*_{vr} \in [0, p_v\phi^*_r],$$
$$L(v, r) < 0 \implies u^*_{vr} = 0.$$

When $\ell = \ell^*_v$, we the LP constraint (2) forces $\sum_r u^*_{vr} = kp_v$.

After setting $u^*_{vr} = p_v\phi^*_r$ for all $L(v, r) > 0$ resources and $u^*_{vr} = 0$ for all $L(v, r) < 0$ resources, let $B_v = kp_v - \sum_{r: L(v,r)>0} p_v\phi^*_r$ be the remaining budget. We distribute B_v across $L(v, r) = 0$ resources by waterfilling (in any order, since each is equally indexed): fill each in turn up to its capacity $p_v\phi^*_r$ until B_v is exhausted. The resulting $(u^*_{vr})_r$ satisfies all three conditions of the waterfill definition with budget kp_v, capacities $(p_v\phi^*_r)_r$, and ordering $r \mapsto L(v, r)$ (with $L > 0$ ranked highest and $L < 0$ ranked lowest). □

4.3 The Fluid Differential Equations

Definition 4.6. For $x \in [0, 1]^{R\times(d+1)}$, let

$$u^{\text{LAG}}_{vr}(x) = p_v \,\text{waterfill}_r(k, (x_{r,0})_r, L(v, \cdot)).$$

In the fluid limit, i.e. as $m, n \to \infty$, the state space is simply $\left\{x \in [0, 1]^{R\times(d+1)} : \forall r, \sum_{i=0}^{d} x_{r,i} = 1\right\}$. We will henceforth use $m, n \to \infty$ to refer to the limit of any sequence $((m_k, n_k))_{k=1}^{\infty}$ where both m_k and n_k are strictly increasing.

Definition 4.7 (Fluid ODE). Define the following state transition map $G : [0, 1]^{R\times(d+1)} \to [0, 1]^{R\times(d+1)}$.

$$G_{r,0}(x) = x_{r,0} - \sum_v u^{\text{LAG}}_{vr}(x) + x_{r,1}$$
$$G_{r,i}(x) = x_{r,i+1} \quad \text{for } i \in \{1, \ldots, d-1\},$$
$$G_{r,d}(x) = \sum_v u^{\text{LAG}}_{vr}(x)$$

The following lemma proves that G describes the behavior of the (m, n)-scaled system as $m, n \to \infty$.

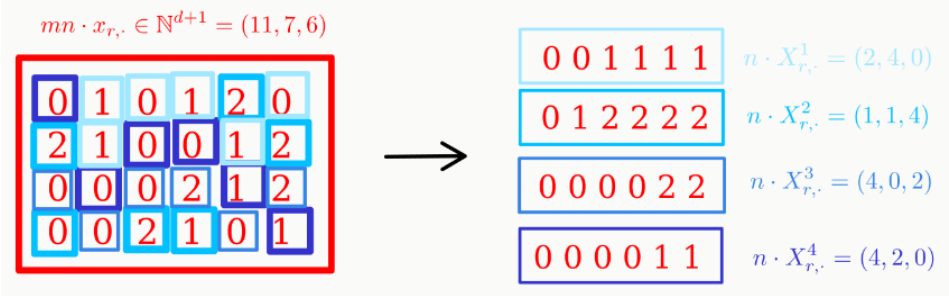


Fig. 4. The mn resources of type r are partitioned uniformly at random into m groups of n. Each resource has state $i \in \{0, \ldots, d\}$. The number of type r state-i resources landing in partition j, denoted $nX^j_{r,i}$, follows a multivariate hypergeometric distribution

Lemma 4.8. *Let $G^{m,n} : \mathbb{P}(\mathcal{X}^{m,n}) \to \mathbb{P}(\mathcal{X}^{m,n})$ denote the transition map of the (m, n)-scaled system. For every $x \in \mathcal{X}^{1,1}$, the random variable $G^{m,n}(x)$ converges in probability to $G(x)$ as $m, n \to \infty$. Further, for every bounded and continuous $f : [0, 1]^{R\times(d+1)} \to \mathbb{R}$, we have*

$$\lim_{m,n\to\infty} \sup_{x\in\mathcal{X}^{1,1}} |\mathbb{E}[f(G^{m,n}(x))] - f(G(x))| = 0.$$

Proof. We examine the transitions from state $x \in \mathcal{X}^{m,n}$ in the (m, n)-scaled system under Lag. Suppose we start in state $x \in \mathcal{X}^{1,1}$. We explicitly describe the state of the (m, n)-scaled system after one timestep, $G^{m,n}(x)$ and compare with the the state of the fluid system after one timestep, $G(x)$. We prove a bound on the gap between the two.

In the (m, n)-scaled system, $x \in [0, 1]^{R\times(d+1)}$ and for each $r \in [R], i \in [0, d]$, $x_{r,i}$ denotes the fraction of type r resources in state i. Note that this fraction is a multiple of $\frac{1}{mn}$, and thus $mn \cdot x_{r,i}$ denotes the count of type r resources in state i.

We begin a timestep by splitting the state counts $mn \cdot x \in \mathbb{N}^{R\times(d+1)}$ into a uniformly drawn partition of m disjoint sets, containing n elements each. An example parititon for fixed r is depicted in Figure 4. This yields for each $r \in [R], j \in [m], n\cdot(X^j_{r,i})^d_{i=0}$ following a multivariate hypergeometric distribution [Siegrist, 2022, Chapter 12.3] with population size mn, type sizes $mn \cdot x_{r,i}$ for each type $i \in [0, d]$ and sample size n. By the properties of the hypergeometric distribution,

$$\mathbb{E}[nX^j_{r,i}] = nx_{r,i}, \qquad \mathrm{Var}(nX^j_{r,i}) = nx_{r,i}(1 - x_{r,i}) \cdot \frac{mn - n}{mn - 1}. \tag{7}$$

Now, for each $j \in [m]$, the j^{th} disjoint set gets a task of type V_j drawn i.i.d. Let nU^j_{vr} denote the number of type (v, r) assignments made by Lag in partition j upon seeing resource states $n \cdot (X^j_{r,i})_{i\in[0,d],r\in[R]}$ and arriving task V_j. Lag's assignment equals the waterfill at those inputs. Thus:

$$U^j_{vr} = \mathbf{1}[V_j = v] \cdot \mathrm{waterfill}_r(k, (X^{(j)}_{r,0})_{r\in[R]}, L(v, \cdot)) \tag{8}$$

Note, a random partition $n \cdot X^{(j)}$ may not have sufficient available resources to meet the partition demand for nk resources, even if the total state $mn \cdot x$ had enough free resources. In these cases, the arrival to the j^{th} partition will be assigned all free resources in $X^{(j)}$ by the waterfill. The total

number of type (v, r) assignments made is $\sum_{j=1}^{m} nU_{vr}^{j}$. After recombining partitions, we have

$$\begin{aligned} G_{r,0}^{m,n}(x) &= x_{r,0} - \frac{1}{m}\sum_{j=1}^{m} U_{vr}^{j} + x_{r,1}, \\ G_{r,i}^{m,n}(x) &= x_{r,i+1} \quad \text{for } i \in \{1, \ldots, d-1\}, \\ G_{r,d}^{m,n}(x) &= \frac{1}{m}\sum_{j=1}^{m} U_{vr}^{j}. \end{aligned}$$

Therefore,

$$\begin{aligned} |G_{r,d}^{m,n}(x) - G_{r,d}(x)| = \Bigg| & \underbrace{\frac{1}{m}\sum_{j=1}^{m}\left(\text{waterfill}_r(k, (X_{r,0}^{(j)})_{r\in[R]}, L(V_j, \cdot)) - \text{waterfill}_r(k, x_{r,0}, L(V_j, \cdot))\right)}_{\text{partition drift}} \\ & + \underbrace{\frac{1}{m}\sum_{j=1}^{m}\left(\text{waterfill}_r(k, x_{r,0}, L(V_j, \cdot)) - \sum_{v} p_v \,\text{waterfill}_r(k, x_{r,0}, L(v, \cdot))\right)}_{\text{arrival drift}}\Bigg| \end{aligned} \tag{9}$$

$$\leq \frac{2}{m}\sum_{j=1}^{m}\sum_{r=1}^{R} |X_{r,0}^{(j)} - x_{r,0}| + \sum_{v} |Q_v - p_v| \,\text{waterfill}_r(k, x_{r,0}, L(v, \cdot)).$$

(where $mQ_v \sim \text{Bin}(m, p_v)$, using Lemma 4.4)

We bound the first summand using Lipschitz continuity of waterfill. After bounding, the first and second summands are independent. The first summand's only source of randomness is the partition drift, while the second summand's is only arrival drift. The partitions and arrivals are drawn independently. Now, by the Hoeffding inequality for $mQ_v \sim \text{Bin}(m, p_v)$, we have

$$\Pr(|Q_v - p_v| > \delta) \leq 2e^{-m\delta^2}.$$

Likewise, using the Serfling inequality [Serfling, 1974] for sampling without replacement we have

$$\Pr\left(|X_{r,0}^{(j)} - x_{r,0}| > \delta\right) \leq 2e^{-n\delta^2}.$$

Substituting these inequalities in (9),

$$\sup_{x \in \mathcal{X}^{m,n}} \Pr[\|G^{m,n}(x) - G(x)\|_\infty > \delta] \leq 2R \cdot 2e^{-n\delta^2} + V \cdot 2e^{-m\delta^2}. \tag{10}$$

Since $[0,1]^{R\times(d+1)}$ is compact and f is continuous, f is uniformly continuous. For any $\epsilon > 0$, choose $\delta > 0$ such that $\|y - z\|_\infty \leq \delta \Rightarrow |f(y) - f(z)| < \epsilon/2$. We bound $|\mathbb{E}[f(G^{m,n}(x))] - f(G(x))|$ as follows: let A be the event that $|G^{m,n}(x) - G(x)| \leq \delta$. Then,

$$\begin{aligned} |\mathbb{E}[f(G^{m,n}(x))] - f(G(x))| &\leq P(A)|\mathbb{E}[f(G^{m,n}(x)) - f(G(x))|A]| \\ &\quad + P(A^c)|\mathbb{E}[f(G^{m,n}(x)) - f(G(x))|A^c]| \\ &\leq \frac{\epsilon}{2} + (2R \cdot 2e^{-n\delta^2} + V \cdot 2e^{-m\delta^2}) \cdot 2\|f\|_\infty. \end{aligned}$$

We use the following bounds in this argument: we bound the first summand using $P(A) \leq 1$, and the uniform continuity bound given $|G^{m,n}(x) - G(x)| \leq \delta$. For the second summand: we use (10)

to bound $P(A^c)$. Further since f is bounded, the difference $|f(G^{m,n}(x)) - f(G(x))|$ is also bounded as $|f(G^{m,n}(x))| + |f(G(x))| \le 2\|f\|_\infty$. Thus, we have that for every ϵ, there exists δ such that

$$\sup_{x \in \mathcal{X}^{1,1}} ||\mathbb{E}[f(G^{m,n}(x))] - f(G(x))| \le \frac{\epsilon}{2} + (2R \cdot 2e^{-n\delta^2} + V \cdot 2e^{-m\delta^2}) \cdot 2\|f\|_\infty.$$

$$\implies \lim_{m,n\to\infty} \sup_{x \in \mathcal{X}^{1,1}} ||\mathbb{E}[f(G^{m,n}(x))] - f(G(x))| \le \frac{\epsilon}{2}.$$

$$\implies \lim_{m,n\to\infty} \sup_{x \in \mathcal{X}^{1,1}} ||\mathbb{E}[f(G^{m,n}(x))] - f(G(x))| = 0.$$

□

Lemma 4.9. *For every m, n, for every initial state $x(0) \in \mathcal{X}^{m,n}$, there exists a stationary distribution $\mu^{m,n} \in \mathbb{P}(\mathcal{X}^{m,n})$ of the (m,n)-scaled system under Lag such that*

$$\frac{\text{Reward}_{m,n}(\textit{LAG})}{mn} = \sum_{v,r} s_{vr} \int u_{vr}^{\textit{LAG}}(x) d\mu^{m,n}(x) + O\left(\frac{1}{\sqrt{n}}\right).$$

Proof. We consider a T-length sample path of the (m,n)-scaled system. Let $X(t)$ be the state at time t and $nU_{vr}^j(t)$ be the number of type (v,r) assignments made by Lag in partition $j \in [m]$ at time t. Let $\text{Reward}(t)$ be the total reward obtained in timestep t. We can write:

$$\frac{\text{Reward}_{m,n}(\text{Lag})}{mn} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \frac{1}{m} \sum_{j=1}^{m} \sum_{v,r} s_{vr}\, U_{vr}^j(t).$$

We next compute $\mathbb{E}[\text{Reward}(t)/mn \mid X(t) = x]$. Given $X(t) = x$, the arrival $V_j \sim (p_v)_v$ is independent of the partition draw $X_{r,i}^{(j)}$. Substituting (8) and taking expectation over V_j:

$$\mathbb{E}\left[U_{vr}^j \mid X(t) = x\right] = p_v \cdot \mathbb{E}\left[\text{waterfill}_r\left(k,\ (X_{r,0}^{(j)})_r,\ L(v,\cdot)\right) \mid X(t) = x\right].$$

All m partitions are drawn by the same procedure, so $X_{r,i}^{(j)} \overset{d}{=} X_{r,i}^{(1)}$ for each j. Summing over j and (v,r):

$$\mathbb{E}\left[\frac{\text{Reward}(t)}{mn} \,\middle|\, X(t) = x\right] = \sum_{v,r} s_{vr}\, p_v\, \mathbb{E}\left[\text{waterfill}_r\left(k,\ (X_{r,0}^{(1)})_r,\ L(v,\cdot)\right) \,\middle|\, X(t) = x\right], \tag{11}$$

where by (7): $\mathbb{E}[X_{r,0}^{(1)}] = x_{r,0}$ and $\text{Var}(X_{r,0}^{(1)}) \le 1/n$.

By Definition 4.6, the gap between the (v,r) term in (11) and $s_{vr} u_{vr}^{\text{Lag}}(x)$ equals

$$s_{vr}\, p_v\, |\mathbb{E}[\text{waterfill}_r(k, (X_{r,0}^{(1)})_r, L(v,\cdot))] - \text{waterfill}_r(k, (x_{r,0})_r, L(v,\cdot))|.$$

We bound this for each (v,r):

$$\left|\mathbb{E}\left[\text{waterfill}_r\left(k, (X_{r,0}^{(1)})_r, L(v,\cdot)\right)\right] - \text{waterfill}_r\left(k, (x_{r,0})_r, L(v,\cdot)\right)\right| \le 2 \sum_{r' \in [R]} \mathbb{E}\left|X_{r',0}^{(1)} - x_{r',0}\right|$$

(by Lemma 4.4)

$$\le 2 \sum_{r' \in [R]} \sqrt{\text{Var}(X_{r',0}^{(1)})} \qquad \text{(Cauchy-Schwarz: } \mathbb{E}|Z| \le \sqrt{\mathbb{E}[Z^2]},\ Z = X_{r',0}^{(1)} - x_{r',0})$$

$$\le \frac{2R}{\sqrt{n}} \qquad \text{(by (7))}$$

Summing over all pairs $(v, r) \in [V] \times [R]$, applying $|s_{vr}| \le s_{\max} := \max_{v,r} |s_{vr}|$, and using $\sum_v p_v = 1$:

$$\left| \mathbb{E}\left[\frac{\text{Reward}(t)}{mn} \,\middle|\, X(t) = x \right] - \sum_{v,r} s_{vr} u_{vr}^{\text{LAG}}(x) \right| \le \sum_{v,r} |s_{vr}|\, p_v \cdot \frac{2R}{\sqrt{n}} \le \frac{2R\, s_{\max}}{\sqrt{n}} \sum_v p_v \cdot R = \frac{2R^2 s_{\max}}{\sqrt{n}}.$$

Setting $C := 2R^2 s_{\max}$, uniformly over all $x \in \mathcal{X}^{m,n}$ and all t:

$$\left| \mathbb{E}\left[\frac{\text{Reward}(t)}{mn} \,\middle|\, X(t) = x \right] - \sum_{v,r} s_{vr}\, u_{vr}^{\text{Lag}}(x) \right| \le \frac{C}{\sqrt{n}}. \tag{12}$$

Taking expectation over (12) and averaging over $t \in [T]$,

$$\left| \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[\frac{\text{Reward}(t)}{mn}] - \frac{1}{T} \sum_{t=1}^{T} \sum_{vr} s_{vr} \mathbb{E}[u_{vr}^{\text{LAG}}(X(t))] \right| \le \frac{C}{\sqrt{n}}.$$

Now further, for every finite DTMC, for every initial state $x(0)$, there exists a stationary distribution μ such that the time average distribution starting from $x(0)$ is μ^{mn}. Thus

$$\lim_{T\to\infty} \mathbb{E}\left[\frac{1}{T} \sum_{t=1}^{T} \sum_{vr} s_{vr} u_{vr}^{\text{LAG}}(x(t)) \right] = \int \sum_{vr} s_{vr} u_{vr}^{\text{LAG}}(x) d\mu^{m,n}(x)$$

Taking $T \to \infty$, we prove that the time average reward exists and is near $\int s_{vr} u_{vr}^{\text{LAG}}(x) d\mu^{mn}(x)$:

$$\left| \frac{\text{Reward}_{m,n}(\text{LAG})}{mn} - \int s_{vr} u_{vr}^{\text{LAG}}(x) d\mu^{mn}(x) \right| \le \frac{C}{\sqrt{n}}.$$

□

4.4 Proof of Fluid Optimality

Assumption 4.10 (Global Attractor Property). *There exists a unique $x^{\lim}$ such that for every initial $x(0)$,*

$$G^t(x(0)) \to x^{\lim} \qquad \text{as } t \to \infty.$$

As noted in [Verloop, 2016], this global attractor property was imposed in [Weber and Weiss, 1990] to establish the asymptotic optimality of Whittle's index policy in a setting with a fixed population of symmetric bandits. In practice, this property is typically difficult to verify analytically and is often established through numerical evidence.

Lemma 4.11. *Given Assumption 4.10, for any sequence $((m_k, n_k))_{k=1}^{\infty}$, the sequence $(\mu^{m_k,n_k})_{k=1}^{\infty}$ converges weakly to δ_{x^*}, the Dirac measure on x^*.*

Proof. We show every sequence $(m_k, n_k) \to \infty$ satisfies $\mu^{m_k,n_k} \Rightarrow \delta_{x^*}$, following the approach of [Benaim and Le Boudec, 2008]. This is equivalent to showing, for every bounded and continuous f,

$$\lim_{m,n\to\infty} \int f(x) d\mu^{m,n} = f(x^*).$$

First, every sequence in $\{\mu^{m,n}\}$ has a convergent subsequence. Each $\mu^{m,n}$ is supported on $\mathcal{X}^{m,n} \subseteq [0,1]^{R\times(d+1)}$. Since $[0,1]^{R\times(d+1)}$ is compact, the family $\{\mu^{m,n}\}_{m,n\ge1}$ is automatically tight. By Prokhorov's theorem [Billingsley, 2013, Theorem 5.1], it is relatively compact and thus every sequence in $\{\mu^{m,n}\}$ has a weakly convergent subsequence with limit μ supported on $[0,1]^{R\times(d+1)}$.

Second, every subsequence limit μ is G-invariant: i.e., for every bounded continuous f, we must have $\int f\,d\mu = \int f(G(x))\,d\mu(x)$. To see this, first, since $\mu^{m_{k_j},n_{k_j}}$ is a stationary distribution under $G^{m_{k_j},n_{k_j}}$:

$$\int f(x)\,d\mu^{m_{k_j},n_{k_j}}(x) \;=\; \int \mathbb{E}\big[f\big(G^{m_{k_j},n_{k_j}}(x)\big)\big]\,d\mu^{m_{k_j},n_{k_j}}(x).$$

We know the LHS converges to $\int f d\mu$ as $m,n\to\infty$. We show the RHS converges to $\int f(G(x))d\mu$:

$$\lim_{m,n\to\infty}\sup_{x\in X^{1,1}}|\mathbb{E}[f(G^{m,n}(x))]-f(G(x))| \;\to\; 0 \qquad \text{(By Lemma 4.8)}$$

$$\implies \lim_{j\to\infty}\int |\mathbb{E}[f(G^{m_{k_j},n_{k_j}}(x))]-f(G(x))|\,d\mu^{m_{k_j},n_{k_j}}(x)\to 0 \qquad \text{(since } f \text{ is bounded)}$$

$$\implies \lim_{j\to\infty}\int \mathbb{E}[f(G^{m_{k_j},n_{k_j}}(x))]d\mu^{m_{k_j},n_{k_j}}(x) = \lim_{j\to\infty}\int f(G(x))d\mu^{m_{k_j},n_{k_j}}(x).$$

Now, the map $x\mapsto f(G(x))$ is bounded and continuous (since G is Lipschitz by Lemma 4.4), and $\mu^{m_{k_j},n_{k_j}}\to\mu$ weakly. Thus:

$$\int f(G(x))\,d\mu^{m_{k_j},n_{k_j}}(x) \;\to\; \int f(G(x))\,d\mu(x).$$

Thus we have $\int f d\mu = \int f(G(x))d\mu$, establishing that μ is G-invariant.

Third, we show that the unique G-invariant measure is δ_{x^*}, and thus μ must equal δ_{x^*}. Let μ be any G-invariant probability measure and f any bounded continuous function. By induction using G-invariance, for all $t\ge 1$:

$$\int f\,d\mu \;=\; \int f(G^t(x))\,d\mu(x).$$

By Assumption 4.10, $G^t(x)\to x^*$ for every $x\in[0,1]^{R\times(d+1)}$, so $f(G^t(x))\to f(x^*)$ pointwise. Since $|f|\le\|f\|_\infty$ uniformly, the dominated convergence theorem gives:

$$\int f\,d\mu \;=\; \lim_{t\to\infty}\int f(G^t(x))\,d\mu(x) \;=\; f(x^*).$$

Since $\int f\,d\mu = f(x^*)$ for all bounded continuous f, we conclude $\mu=\delta_{x^*}$.

□

Theorem 4.12 (Asymptotic optimality of Lag). *Given Assumption 4.10,*

$$\frac{\text{Reward}_{m,n}(\textit{LAG})}{mn} \;\to\; \text{Reward}^*(\textit{OCC-LP}) \qquad \textit{as } m,n\to\infty.$$

Proof. We consider the sequence of distributions $\{\mu^{m,n}\}_{m,n\in\mathbb{N}}$ provided by Lemma 4.9. By Lemma 4.11, we know that every sequence $((m_k,n_k))_{k=1}^{\infty}$ converges weakly to δ_{x^*}. By Lemma 4.4, we have that $x\mapsto u_{vr}^{\text{LAG}}(x)$ is bounded and continuous, thus the map $\mu\mapsto\int u_{vr}^{\text{LAG}}(x)d\mu$ and further, $\mu\mapsto\sum_{vr}s_{vr}\int u_{vr}^{\text{LAG}}(x)d\mu$ are continuous with respect to weak convergence.

Therefore,

$$\begin{aligned}\frac{\text{Reward}_{m,n}(\textsc{Lag})}{mn} &= \sum_{v,r} s_{vr} \int u_{vr}^{\textsc{Lag}}(x) d\mu^{m,n}(x) + O\left(\frac{1}{\sqrt{n}}\right) \\ &\to \sum_{vr} s_{vr} \int u_{vr}^{\textsc{Lag}}(x) d\delta_{x^*} \text{ as } m, n \to \infty \\ &= \sum_{vr} s_{vr} u_{vr}^{\textsc{Lag}}(x^*) \\ &= \sum_{vr} s_{vr} u_{vr}^{*} \\ &= \text{Reward}^*(OCC-LP).\end{aligned}$$

□

5 Experiments

We evaluate the performance of our proposed policies from Sections 3.2 and 3.3, comparing it against other policies in the literature. In our evaluations, we experiment with real datasets from journal peer review and ride sharing, synthetic instances and a range of system loads. We present the following main findings:

- Across all experiments, our Lag policy consistently obtains the highest, or within 1% of highest, average match score per arrival among all candidate policies we considered. It consistently obtains 94% to 99% of the OCC-LP upper bound. Whi also matches its performance in all but one synthetic instance which we discuss in detail.
- On real journal review data, Lag and Whi both achieve 20% improvement over the baseline of Greedy paper-reviewer assignment.
- In certain synthetic instances, both Lag and Whi outperform other candidate policies by 10-66%.

5.1 Policies Evaluated

We compare our policy with a broad spectrum of alternative policies from the prior literature, as well as theoretical upper bounds:

Index Policies (ours).

Lag: Derived from the Lagrangian relaxation of the OCC-LP (Section 3.2). The index

$$L(v,r) = s_{vr} - \ell_v^* - d \cdot q_r^*$$

incorporates dual variables ℓ_v^* (task-side) and q_r^* (resource-side) from solving a one-time offline LP. When a type v task arrives, we assign the k available resources with highest index $L(v,\cdot)$.

Whi: Derived using the classic RMAB "subsidy of indifference" approach (Section 3.3). The closed-form index

$$W(v,r) = s_{vr} - d\sum_{v'} p_{v'} \max(s_{v'r} - s_{vr}, 0).$$

When a type v task arrives, we assign the k available resources with highest index $W(v,\cdot)$.

Baselines.

Greedy: Selects the k available resources with highest current scores, ignoring future opportunity costs entirely.

Random: Selects k available resources uniformly at random, ignoring scores entirely.

ALG-SC-LP: The safe-choice LP policy of [Dickerson et al., 2021]. ALG-SC-LP solves the KIID LP ([Dickerson et al., 2021, LP (1)]) offline and obtains optimal allocation solution $x^* \in \mathbb{R}^{V \times R}$. At each timestep, it selects resource $r \in \mathcal{A}_t$ with probability $x^*_{vr} / \sum_{r' \in \mathcal{A}_t} x^*_{vr'}$. It defaults to Greedy if no available resource has positive LP weight.

Upper Bounds. The following LP optimal values are not online policies; they are shown as reference lines in all figures.

OCC-UB: Optimal value of our occupancy LP (Section 3.1), an upper bound for any online policy. Tighter than KIID-UB by construction.

KIID-UB: Optimal value of the KIID LP of [Dickerson et al., 2021], a coarser upper bound.

5.2 Performance with Load

The system "load" can be characterized as $\rho = kd/R$. This is because we have one job arriving each time step, and each job demands k resources for d timesteps each. We have R servers available to serve arriving tasks.

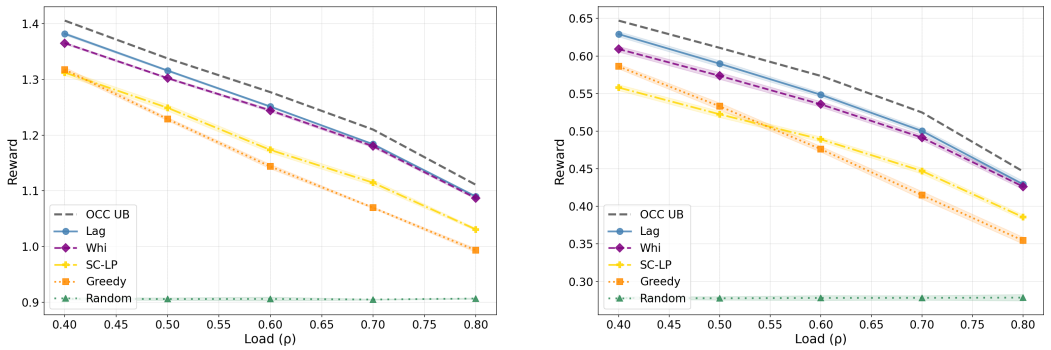
In the following experiments, we fix k and vary the delay $d \in [0, \frac{R}{k}]$, parameterizing system load $\rho = kd/R \in (0, 1)$. For each load point we re-solve the LP and recompute all indices; the simulation runs 5 trials of $T = 5{,}000$ steps. We compare five policies: Lag, Whi, ALG-SC-LP, Greedy, and Random as described in Section 5.1. We test five instances. Three experiments using real datasets: from journal peer-review assignment (ICLR and TMLR instances in Table 2, Figure 5) and the specialist-generalist failure mode of Greedy (Specialist instance, Figure 6). Two experiments use synthetic instances which demonstrate different performance regimes (Unfriendly and Low-rank instances, Figure 7). Table 2 provides a summary of the relative performance of all policies evaluated on all instances.

Table 2. Reward as % of OCC-UB at load $\rho = 0.7$.

Instance	Lag	Whi	ALG-SC-LP	Greedy	Random
ICLR	97.8	97.5	92.1	88.4	74.8
TMLR	95.3	93.6	85.2	79.0	53.0
Specialist	96.4	95.4	90.6	83.6	59.4
Unfriendly	95.0	95.0	86.4	57.2	45.4
Low-rank	96.3	79.2	94.7	88.9	66.7

In Figure 5, we present two problem instances derived from peer-review assignment data, to model online paper-reviewer assignment for peer-review journals. In both experiments, the arriving tasks represent submitted papers, and the resource pool represents available reviewers. The similarity scores s_{vr} are computed using term frequency-inverse document frequency (tf-idf) [Manning, 2008]: an NLP-based computation of the similarity between the submitted paper and each reviewer's recent papers. For Figure 5a, we use the ICLR conference similarity scores provided by [Xu et al., 2019, Section A.1]. For Figure 5b, we compute TMLR journal similarity scores using a similar approach. We use R=30 and subsample V=10 task types uniformly at random from the full set observed in the data. We use $k = 3$ reviewer assignments per submitted paper.

In Figure 5a, all policies except Random closely track OCC-UB. While in Figure 5b, there is more performance variation among the policies. In both settings, Lag performs best, followed by Whi, ALG-SC-LP, Greedy and Random in order. At $\rho = 0.7$, Lag achieves 97.8% of the OCC-UB on ICLR



(a) ICLR ($R = 30$, $V = 10$, $k = 3$) (b) TMLR ($R = 30$, $V = 10$, $k = 3$)

Fig. 5. Reward vs. delay d on real reviewer-paper datasets. LAG tracks the OCC-UB across all loads; Greedy and ALG-SC-LP degrade at high load.

and 95.3% on TMLR, while Greedy reaches only 88% and 79% respectively. Thus notably, LAG and WHI achieve a 20% improvement in assignment scores over Greedy, the widely used baseline for journal reviewer assignments, on the TMLR similarity score dataset.

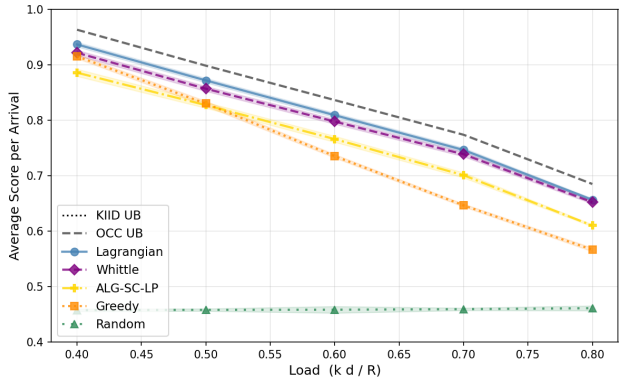
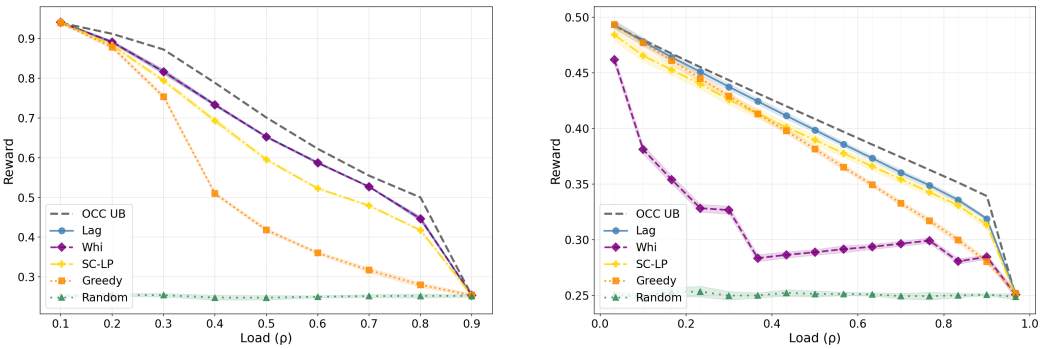


Fig. 6. Reward vs. delay d on the Specialist instance ($R = 30$, $V = 10$, $k = 3$).

In Figure 6, we present the Specialist instance, constructed from the TMLR dataset to expose the specialist-generalist failure mode of Greedy [Zhao and Zhang, 2022]. We classify reviewers into four archetypes based on per-reviewer statistics: *veterans* (high mean and high maximum score); *specialists* (high maximum but low mean); *generalists* (low coefficient of variation and moderate mean); and *novices* (low mean and low maximum). As in the ICLR and TMLR experiments, we use R=30 reviewers, V=10 paper types subsampled uniformly at random, and $k = 3$ assignments per paper.

The failure mode arises because veterans outscore specialists on most paper types, so Greedy assigns veterans first whenever one is available. Under high load, veterans become scarce; papers that a specialist handles best are instead assigned a generalist, since Greedy picks the highest-scoring available reviewer without reserving capacity. In Figure 6, Greedy falls steeply with load, reaching only 83.6% of OCC-UB at $\rho = 0.7$, while LAG attains 96.4%—a gap of 12.8%. ALG-SC-LP

(90.6%) improves over Greedy but remains well below Lag. Whi tracks Lag closely, staying within 1 pp across all loads. KIID-UB is so loose on this instance that it falls entirely outside the plot range.



(a) Unfriendly ($R = 10$, $V = 32$, $k = 1$)

(b) Low-rank ($R = 30$, $V = 10$, $k = 1$)

Fig. 7. Reward vs. delay d on synthetic instances. On the low-rank instance Whi falls substantially below Greedy; Lag stays near the OCC-UB in both cases.

In Figure 7, we present two synthetic instances designed to expose complementary failure modes of the policies.

In Figure 7a, the Unfriendly instance ($R = 10$, $V = 32$) has $g = 5$ good resource types and $b = 5$ "dummy" resource types. Dummy resources score 0 on every task type and exist solely as a capacity buffer. Each good r resource has two score levels: a high score ($s_{vr} = 1.0$) for tasks it is well-matched to, and a near-zero score ($s_{vr} = 0.01$) for all others; task types arrive uniformly. The near-zero score is strictly positive, so Greedy assigns a good resource even when it is a poor match—wasting it on a task which could be handed to a dummy resource—rather than holding it for a future high-reward arrival. The gap between Greedy and OCC-UB can be made arbitrarily large by varying the problem parameters. In Figure 7a and Table 2, Greedy achieves only 57% of the OCC-UB at $\rho = 0.7$, whereas Lag attains 95%. Whi coincides with Lag on this instance: since both policies capture the desired behavior of "saving" good resources for the highly compatible tasks.

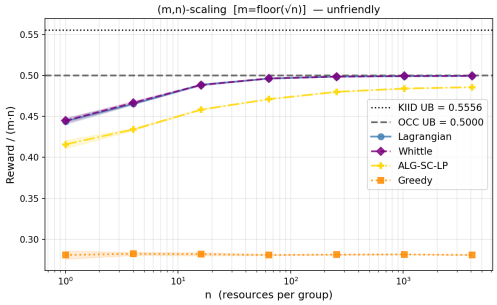
In Figure 7b, the LowRank instance is designed with highly correlated scores across resources. In particular, it uses a rank-1 similarity score matrix $s_{vr} = a_v b_r$, where $a_v = v/(V-1)$ and $b_r = r/(R-1)$ are linearly spaced in $[0, 1]$. Task types again arrive uniformly. In Figure 7b, we present instances with $R = 30, V = 10, k = 1$. In this setting, Lag performs best, followed by ALG-SC-LP, Greedy, Whi and Random in order. At $\rho = 0.7$, Whi falls to 79% of the OCC-UB and 11% *below* Greedy, while Lag maintains 96% of the OCC-UB, and is 8% better than Greedy. Figure 7b reveals an interesting phenomenon: In this setting, the Whittle index becomes

$$W(v, r) = b_r \left(a_v - \frac{d}{2}(1 - a_v)^2 \right).$$

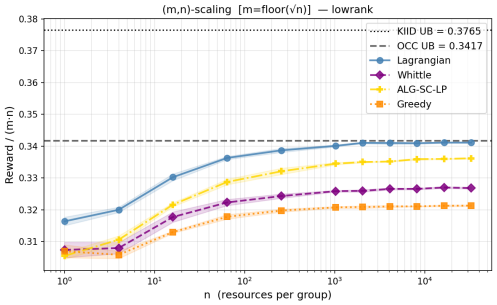
Namely, there is a threshold τ_d such that when a low-quality task arrives (with $a_v < \tau_d$), Whittle assigns anti-greedily: assigning the worst (lowest b_r) available resource. However, this opportunity cost does not pay off in this distribution, as the future value of preserving resources is overestimated. While this approach seems reasonable, assigning greedily to every arriving task achieves considerably better assignment scores.

5.3 Mean field limit

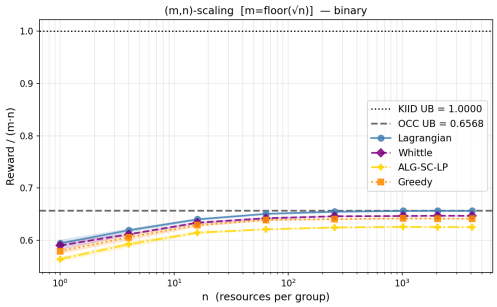
We study the empirical convergence of the (m, n)-scaled system's expected reward per arrival to the value of OCC-LP as $m, n \to \infty$. In each figure, we increase m, n in tandem as $m = \sqrt{n}$, though we have verified that other increasing sequences $(m_k, n_k)_k$ also converge. We plot n in the x-axis and average reward per arrival as the y-axis. Each data point is the mean of 5 independent trials, each run for $T = 5{,}000$ timesteps; error bars show 95% confidence intervals ($\pm 1.96\,\hat{\sigma}/\sqrt{5}$). We compare Lag, Whi, ALG-SC-LP, and Greedy (Section 5.1) and display OCC-UB and KIID-UB as horizontal reference lines. We test four instances spanning synthetic and real-data regimes: Unfriendly, Low-rank, Binary, and TMLR.



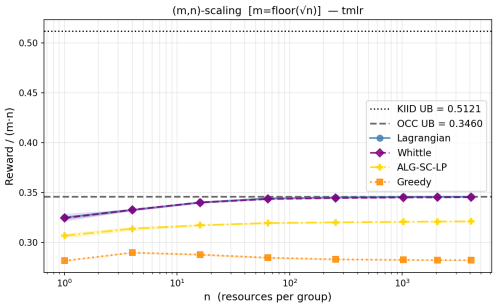
(a) Unfriendly ($R = 10$, $V = 32$, $k = 1$)



(b) Low-rank ($R = 10$, $V = 5$, $k = 1$)



(c) Binary ($R = 10$, $V = 10$, $k = 1$)



(d) TMLR ($R = 10$, $V = 10$, $k = 1$)

Fig. 8. Reward per copy (reward/(mn)) vs. n under the (m, n)-scaled system with $m = \lfloor \sqrt{n} \rfloor$. Lag converges to OCC-UB across all four instances, confirming Theorem 4.12.

In Figure 8a, we present the Unfriendly instance ($R = 10$, $V = 32$, $k = 1$, $d = 8$), as described in Section 5.2: five good resource types and five dummy resource types; each good resource r scores $s_{vr} = 1.0$ if bit r of task type v is set and $s_{vr} = 0.01$ otherwise; dummy resources score 0; task types arrive uniformly ($p_v = 1/32$). The system load is $\rho = kd/R = 0.8$, with OCC-UB = 0.500 and KIID-UB = 0.556. As n increases, Lag and Whi both converge to the OCC-UB, reaching 99.9% of OCC-UB at $n = 4096$ (0.4995 and 0.4994, respectively). ALG-SC-LP improves with n but converges to a lower plateau (97.1% of OCC-UB at $n = 4096$). Greedy remains flat at ≈ 0.281 across all n (56.2% of OCC-UB).

In Figure 8b, we present the Low-rank instance ($R = 10$, $V = 5$, $k = 1$, $d = 8$). As in Section 5.2, the similarity matrix is rank-1:

$$s_{vr} = a_v\, b_r, \qquad a_v = \tfrac{v}{V-1}, \quad b_r = \tfrac{r}{R-1}, \tag{13}$$

with a_v, b_r linearly spaced in $[0, 1]$, task types arriving uniformly ($p_v = 1/5$), and load $\rho = kd/R = 0.8$. OCC-UB = 0.342 and KIID-UB = 0.377. Lag converges to the OCC-UB, reaching 99.8% at $n = 4096$ (0.341). ALG-SC-LP is second (98.1% of OCC-UB at $n = 4096$) but does not converge to OCC-UB within the plotted range. Whi and Greedy trail further behind, plateauing at 95.6% and 94.0% of OCC-UB at $n = 4096$, respectively; the ordering is Lag > ALG-SC-LP > Whi > Greedy.

In Figure 8c, we present the Binary instance ($R = 10$, $V = 10$, $k = 1$, $d = 8$). The similarity matrix is drawn once as

$$s_{vr} \sim \text{Bernoulli}(0.3), \qquad v \in [V],\ r \in [R], \tag{14}$$

using random seed 0, and held fixed across all trials; task types arrive uniformly ($p_v = 1/10$). The load is $\rho = kd/R = 0.8$, with OCC-UB = 0.657 and KIID-UB = 1.000 (the KIID relaxation is vacuously loose on this instance). Lag converges to OCC-UB, reaching 99.9% at $n = 4096$ (0.6564). Whi and Greedy both improve monotonically with n, reaching 98.5% (0.6468) and 97.7% (0.6414) of OCC-UB at $n = 4096$, respectively. ALG-SC-LP performs worst among the four policies, plateauing at 95.2% (0.6253) of OCC-UB; the ordering is Lag > Whi > Greedy > ALG-SC-LP.

In Figure 8d, we present the TMLR instance ($R = 10$, $V = 10$, $k = 1$, $d = 8$), using a random subsample of 10 reviewers and 10 paper types from the TMLR similarity matrix (809 reviewers × 418 papers, normalized to $[0, 1]$; seed 0), as described in Section 5.2. Task types arrive uniformly ($p_v = 1/10$); load $\rho = kd/R = 0.8$; OCC-UB = 0.346 and KIID-UB = 0.512. Both Lag and Whi converge to the OCC-UB, reaching 99.97% (0.3459) and 99.9% (0.3455) at $n = 4096$, respectively. ALG-SC-LP improves with n but converges to a lower plateau (92.8% of OCC-UB at $n = 4096$). Greedy remains substantially below the upper bound (81.6% of OCC-UB at $n = 4096$). The ordering Lag ≈ Whi ≫ ALG-SC-LP ≫ Greedy mirrors the pattern observed on the Unfriendly instance.

6 Conclusion

We studied online task assignment with reusable resources, formulated as a contextual restless multi-armed bandit, and derived an occupancy-measure LP upper bound on any online policy's performance. Our main contribution is the Lag policy, which is asymptotically optimal in the (m,n)-scaling regime where both arrivals and resource copies grow. We also developed the Whittle index, which has a natural interpretation: both policies offset the raw match score by an opportunity cost term reflecting the value of keeping a resource available for future assignments. Empirically, Lag achieves 94-99% of the LP upper bound and substantially outperforms greedy baselines, improving by around 20% on reviewer data.

References

Konstantin Avrachenkov, Vivek S Borkar, and Pratik Shah. 2026. Lagrangian index policy for restless bandits with average reward. *Queueing Systems* 110, 1 (2026), 21.

Michel Benaim and Jean-Yves Le Boudec. 2008. A class of mean field interaction models for computer and communication systems. *Performance evaluation* 65, 11-12 (2008), 823–838.

Patrick Billingsley. 2013. *Convergence of probability measures.* John Wiley & Sons.

Humayun Kabir Biswas and Md Maruf Hasan. 2007. Using Publications and Domain Knowledge to Build Research Profiles: An Application in Automatic Reviewer Assignment. In *2007 International Conference on Information and Communication Technology*. 82–86. https://doi.org/10.1109/ICICT.2007.375347

Laurent Charlin and Richard Zemel. 2013. The Toronto paper matching system: an automated paper-reviewer assignment system. (2013).

Xin Chen and I-Hong Hou. 2024. Contextual restless multi-armed bandits with application to demand response decision-making. In *2024 IEEE 63rd Conference on Decision and Control (CDC)*. IEEE, 2652–2657.

Steven Delong, Alireza Farhadi, Rad Niazadeh, Balasubramanian Sivan, and Rajan Udwani. 2024. Online Bipartite Matching with Reusable Resources. *Mathematics of Operations Research* 49, 3 (2024), 1825–1854. https://doi.org/10.1287/moor.2022.0242 Published online: October 13, 2023.

John P. Dickerson, Karthik A. Sankararaman, Aravind Srinivasan, and Pan Xu. 2021. Allocation Problems in Ride-sharing Platforms: Online Matching with Offline Reusable Resources. *ACM Transactions on Economics and Computation* 9, 3 (September 2021), 1–17. https://doi.org/10.1145/3456756

Matthew Fahrbach, Zhiyi Huang, Runzhou Tao, and Morteza Zadimoghaddam. 2022. Edge-Weighted Online Bipartite Matching. *J. ACM* 69, 6 (November 2022), 1–35. https://doi.org/10.1145/3556971 FOCS 2020 Best Paper.

Hui Fang and ChengXiang Zhai. 2007. Probabilistic Models for Expert Finding. In *Advances in Information Retrieval*, Giambattista Amati, Claudio Carpineto, and Giovanni Romano (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 418–430.

Jon Feldman, Aranyak Mehta, Vahab Mirrokni, and Shan Muthukrishnan. 2009. Online stochastic matching: Beating 1-1/e. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*. IEEE, 117–126.

Xiao-Yue Gong, Vineet Goyal, Garud N. Iyengar, David Simchi-Levi, Rajan Udwani, and Shuangyu Wang. 2022. Online Assortment Optimization with Reusable Resources. *Management Science* 68, 7 (2022), 4772–4785. https://doi.org/10.1287/mnsc.2021.4134

Zhiyi Huang, Zhihao Gavin Tang, and David Wajc. 2024. Online Matching: A Brief Survey. *ACM SIGecom Exchanges* 22, 1 (October 2024), 135–158. https://doi.org/10.1145/3699824.3699837

Richard M. Karp, Umesh V. Vazirani, and Vijay V. Vazirani. 1990. An Optimal Algorithm for On-line Bipartite Matching. In *Proceedings of the Twenty-second Annual ACM Symposium on Theory of Computing (STOC '90)*. ACM, New York, NY, USA, 352–358. https://doi.org/10.1145/100216.100262

Maialen Larrañaga. 2015. *Dynamic control of stochastic and fluid resource-sharing systems*. Ph. D. Dissertation. Institut National Polytechnique de Toulouse-INPT; Universidad del País Vasco.

Christopher D Manning. 2008. *Introduction to information retrieval*. Syngress Publishing,.

Vahideh H Manshadi, Shayan Oveis Gharan, and Amin Saberi. 2012. Online stochastic matching: Online actions based on offline statistics. *Mathematics of Operations Research* 37, 4 (2012), 559–573.

Vedant Nanda, Pan Xu, Karthik Abhinav Sankararaman, John Dickerson, and Aravind Srinivasan. 2020. Balancing the tradeoff between profit and fairness in rideshare platforms during high-demand hours. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 2210–2217.

José Niño-Mora. 2023. Markovian restless bandits and index policies: A review. *Mathematics* 11, 7 (2023), 1639.

Christos H Papadimitriou and John N Tsitsiklis. 1987. The complexity of Markov decision processes. *Mathematics of operations research* 12, 3 (1987), 441–450.

Robert J Serfling. 1974. Probability inequalities for the sum in sampling without replacement. *The Annals of Statistics* (1974), 39–48.

M. Shanks, G. Yu, and S. H. Jacobson. 2023. Approximation Algorithms for Stochastic Online Matching with Reusable Resources. *Mathematical Methods of Operations Research* 98 (2023), 43–56. https://doi.org/10.1007/s00186-023-00822-3

Kyle Siegrist. 2022. The Multivariate Hypergeometric Distribution. Section 12.3. License: CC BY 2.0. Accessed: April 2026.

Ivan Stelmakh, Nihar Shah, and Aarti Singh. 2021. PeerReview4All: Fair and accurate reviewer assignment in peer review. *Journal of Machine Learning Research* 22, 163 (2021), 1–66.

Hanna Sumita, Shinji Ito, Kei Takemura, Daisuke Hatano, Takuro Fukunaga, Naonori Kakimura, and Ken-ichi Kawarabayashi. 2022. Online Task Assignment Problems with Reusable Resources. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 5199–5207. https://doi.org/10.1609/aaai.v36i5.20455

I. M. Verloop. 2016. ASYMPTOTICALLY OPTIMAL PRIORITY POLICIES FOR INDEXABLE AND NONINDEXABLE RESTLESS BANDITS. *The Annals of Applied Probability* 26, 4 (2016), 1947–1995. http://www.jstor.org/stable/24810047

Richard R Weber and Gideon Weiss. 1990. On an index policy for restless bandits. *Journal of applied probability* 27, 3 (1990), 637–648.

Peter Whittle. 1988. Restless Bandits: Activity Allocation in a Changing World. *Journal of Applied Probability* 25, A (1988), 287–298. https://doi.org/10.2307/3214163

Yichong Xu, Han Zhao, Xiaofei Shi, and Nihar B Shah. 2019. On strategyproof conference peer review. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 616–622.

Xiquan Zhao and Yangsen Zhang. 2022. Reviewer assignment algorithms for peer review automation: A survey. *Information Processing & Management* 59, 5 (2022), 103028. https://doi.org/10.1016/j.ipm.2022.103028