Adaptive Matching for Expert Systems with Uncertain Task Types

VIRAG SHAH, LENNART GULIKERS, and LAURENT MASSOULIÉ, Microsoft Research-INRIA Joint Centre
MILAN VOJNOVIĆ, London School of Economics

Abstract

Online two-sided matching markets such as Q&A forums (e.g. StackOverflow, Quora) and online labour platforms (e.g. Upwork) critically rely on the ability to propose adequate matches based on imperfect knowledge of the two parties to be matched. This prompts the following question: Which matching recommendation algorithms can, in the presence of such uncertainty, lead to efficient platform operation?

To answer this question, we develop a model of a task / server matching system. For this model, we give a necessary and sufficient condition for an incoming stream of tasks to be manageable by the system. We further identify a so-called back-pressure policy under which the throughput that the system can handle is optimized. We show that this policy achieves strictly larger throughput than a natural greedy policy. Finally, we validate our model and confirm our theoretical findings with experiments based on logs of Math.StackExchange, a StackOverflow forum dedicated to mathematics.

1 INTRODUCTION

Online platforms that enable matches between trading partners in two-sided markets have recently blossomed in many areas: LinkedIn and Upwork facilitate matches between employers and employees; Uber allows matches between passengers and car drivers; Airbnb and Booking.com connect travelers and housing facilities; Quora and Stack Exchange facilitate matches between questions and either answers, or experts able to provide them.

All these systems crucially rely on the ability to propose adequate matches based on imperfect knowledge of the characteristics of the two parties to be matched. For example, in the context of online labour platforms, there is uncertainty about both the skill sets of candidate employees and the job requirements. Similarly, in the context of online Q&A platforms, there is uncertainty about both question types and users' ability to provide answers.

This naturally leads to the following question: *which matching recommendation algorithms can, in the presence of such uncertainty, lead to efficient platform operation?* A natural measure of efficiency is the throughput that the platform achieves, i.e. the rate of successful matches it allows. To address this question, one thus needs first to characterize fundamental limits on the achievable throughput.

In this paper, we progress towards answering these questions as follows.

First, we propose a simple model of such platforms, which features a static collection of servers, or experts on the one hand, and a continuous stream of arrivals of tasks, or jobs, on the other hand. In our model, the platform's operation consists of servers iteratively attempting to solve tasks. After being processed by some server, a task leaves the system if solved; otherwise it remains till successfully treated by some server. To model uncertainty about task types, we assume that for each incoming task we are given the prior distribution of this task's "true type". Servers' abilities are then represented via the probability that each server has to solve a task of given type after one attempt at it.

In a Q&A platform scenario, tasks are questions, and servers are experts; a server processing a task corresponds to an expert providing an answer to a question. A task being solved corresponds to an answer being accepted. In an online labour platform, tasks could be job offers, and a server may be a pool of workers with similar abilities. A server processing a task then corresponds to a worker being interviewed for a job, and the task is solved if the interview leads to a hire. We could also consider the dual interpretation when the labour market is constrained by workers rather than job offers. Then a task is a worker seeking work, while a server is a pool of employers looking for hires.

An important feature of our model consists in the fact that when a task's processing does not lead to failure, it does however affect uncertainty about the task's type. Indeed, the a posteriori distribution of the task's type after a failed attempt on it by some server differs from its prior distribution. For instance in a Q&A scenario, a question which an expert in Calculus failed to answer either is not about Calculus, or is very hard.

For our model, we then determine necessary and sufficient conditions for an incoming stream of task arrivals to be manageable by the servers, or in other words, determine achievable throughputs of the system. In the process we introduce candidate policies, in particular the greedy policy according to which a server choses to serve tasks for which its chance of success is highest. This scheduling strategy is both easy to implement and is based on a natural motivation. Surprisingly perhaps, we show that it is not optimal in the throughput it can handle. In contrast, we introduce a so-called backpressure policy inspired from the wireless networking literature [33], which we prove to be throughput-optimal.

To validate our modeling and theoretical results, we analyze logs from Math.StackExchange, a StackOverflow forum specializing on mathematics. Our data analysis corroborates several aspects of our model, notably the modeling of uncertainty about task types as a prior on task type, together with the modeling of server ability through their success probability on each task type. Numerical experiments based on Math.StackExchange data analysis confirm the theoretical results, namely that our backpressure policy can achieve significantly higher throughputs than other baseline policies.

We summarize contributions of this paper as follows:

- We propose a new model of a generic task-expert system that allows for uncertainty of task types, heterogeneity of skills, and recurring attempts of experts in solving tasks.
- We provide a full characterization of the stability region, or sustainable throughputs, of the task-expert system under consideration. We establish that a particular backpressure policy is throughput-optimal, in the sense that it supports maximum task arrival rate under which the system is stable.
- We show that there exist instances of task-expert systems under which simple matching policies such as a natural greedy policy and a random policy can only support a much smaller maximum task arrival rate, than the backpressure policy.
- We report the results of empirical analysis of the popular Math.StackExchange Q&A platform which establish heterogeneity of skills of experts, with experts knowledgeable across different types of tasks and others specialized in particular types of tasks. We also show numerical evaluation results that confirm the benefits of the backpressure policy on greedy and random matchmaking policies.

The remainder of the paper is structured as follows. Section 2 presents our system model. In Section 3, we present results for two baseline matchmaking policies, namely Greedy and Random, and a characterization of achievable throughputs under Random. Section 4 presents the characterization of task arrival rates that can be supported under which the system is stable and prove the superiority of backpressure policy over Random and Greedy. In Section 5, we present our experimental results. Related work is discussed in Section 6. We conclude in Section 7. Proofs of the results are provided in the Section 8.

2 SYSTEM MODEL

We denote by C the set of types that tasks can take, and assume $|C| < +\infty$. We let $\mathcal{C}$ denote the set of probability distributions on C. We assume that each incoming task has an associated type which is not observed. Instead, the probability distribution $z = \{z_c\}_{c\in C} \in \mathcal{C}$ of the task's type is provided. Throughout, we will refer to such distributions as *mixed types*. The rationale for assuming that such mixed types are available is that they represent side information revealed upon task arrival (e.g. for a question in a Q&A forum, side information could be its text and associated tags; in online labor platform this could be a job description and prior knowledge about the company posting the offer).

We let S be a set of servers (or experts) present in the system. For each $s \in S$, $c \in C$, $p_{s,c}$ denotes the probability that server s solves a task of type c after processing it. We assume that a given task may be inspected several times by a given server and that the outcomes success / failure are independent at each inspection. This can be justified if a label s in fact represents a collection of experts with similar abilities, in which case multiple processings by s correspond to processing by distinct individual experts.

We make the following statistical assumptions. The subsequent mixed types Z_i of incoming tasks are assumed i.i.d., taking values in a countable subset $\mathcal{Z}$ of $\mathcal{C}$, and we let $\pi_z = P(Z_i = z)$ for all $z \in \mathcal{Z}$. Tasks arrive at a rate of λ per time unit on average. Finally, the time for server s to complete an attempt on a job takes on average $1/\mu_s$ time units, and such attempt durations are i.i.d.. All involved sources of randomness are independent. For simplicity we assume more specifically that tasks arrive at the instants of a Poisson process with intensity λ, and that the time for server s to complete an attempt at a task follows an Exponential distribution with parameter μ_s. These assumptions will imply that the system state at any given time t can be represented as a Markov process, which simplifies analysis, but is not essential for the throughput optimality properties that concern us here.

We now describe direct consequences of the assumptions just made. When server s processes a task with mixed type $z \in \mathcal{Z}$, then the probability that it fails is given by

$$\psi_s(z) = \sum_{c \in C} z_c(1 - p_{s,c}). \tag{1}$$

Moreover, conditional on such failure, Bayes formula readily implies that the mixed type of the job then becomes

$$\phi_s(z) = \left\{ \frac{z_c(1 - p_{s,c})}{\psi_s(z)} \right\}_{c \in C} \tag{2}$$

These two functions will be instrumental in the sequel to describe policies of interest. They will also be used to characterize arrival rates λ for which the system can be *stabilized*, i.e. for which there exists a scheduling policy which induces a stationary regime of the system's behavior. This is our primary concern in this work: indeed for a stable system the long term task resolution rate coincides with the task arrival rate λ, and thus throughput-optimal policies must make the system stable whenever this is possible.

We close the section with additional assumptions and notations.

We assume that $\mathcal{Z}$ is closed under $\phi_s(\cdot)$, i.e., for each $z \in \mathcal{Z}$, $\phi_s(z) \in \mathcal{Z}$. This loses no generality, as the closure of a countable set with respect to a finite number of maps ϕ_s remains countable. At any time t let $N_z(t)$ represent the number of tasks of mixed-type z present in the system and $N(t) = \{N_z(t)\}_{z \in \mathcal{Z}}$. We also let $z(s,t)$ denote the mixed type of that task that server s works on at time t. For strategies such that the servers select which task to handle based uniquely on the vector $N(t)$, the process $\{N(t)\}_{t \geq 0}$ forms a continuous-time Markov chain (CTMC) [6]. The policies considered in this paper are studied by analyzing the associated CTMC.

We allow a task to be assigned to multiple experts at a given time. Further, we allow pre-emptive service, i.e., an expert may drop service of a task should a new task arrive into a system or an existing task receive a response.

3 BASELINE POLICIES

We consider a natural Greedy policy where each expert is assigned a task which best suits its skills, i.e., among the outstanding tasks, an expert s is assigned a task of a mixed-type z which minimizes $\psi_s(z)$. A question arises: does a myopic Greedy policy optimize the long term throughput in a stabilizable system? We will show via an example that the answer to this question is negative. We also consider a Random policy, where each expert is assigned a task chosen uniformly at random and characterize its stability region. We provide example scenarios where both Greedy and Random have stability regions which are equivalent, and yet strictly sub-optimal.

3.1 Greedy Policy

Definition 3.1 (Greedy Policy). A policy is Greedy if given the system state each expert is assigned an outstanding task which maximizes its success probability, i.e., for each N such that $|N| > 0$ we have

$$z(s) \in A_s(N) = \arg \min_{z: N_z > 0} \psi_s(z),$$

where ties could be broken arbitrarily, e.g., uniformly at random among this set.

If ties are broken uniformly at random then the transition rates for the CTMC under greedy policy are given as follows. Let $q(n, n')$ be the transition rate from state n to state n'. Let e_z denote the vector with all coordinates equal to 0 except for the z-coordinate which equals 1. Fix a state n. For each $z \in \mathcal{Z}$ we have

$$\begin{aligned} q(n, n + e_z) &= \lambda \pi_z, \\ q(n, n - e_z) &= \sum_{s: z \in A_s(N)} \mu_s (1 - \psi_s(z)) \frac{1}{|A_s(N)|}, \\ q(n, n - e_z + e_{\phi_s(z)}) &= \sum_{s: z \in A_s(N)} \mu_s \psi_s(z) \frac{1}{|A_s(N)|}. \end{aligned}$$

Transition rate $q(n, n')$ for each (n, n') which is not as given above is equal to 0.

We will consider the following task-expert system instance.

Definition 3.2 (Two types-two experts system). Suppose that there are two task types $C = \{c_1, c_2\}$ and two experts $S = \{s_1, s_2\}$. Each arrival is equally likely to be of both types, i.e., $\pi_{z'} = 1$ where z' satisfies $z'_c = 1/2$ for each $c \in C$, and $\pi_z = 0$ if $z \neq z'$. Both experts provide responses at unite rate, i.e., $\mu_s = 1$ for each s. Further, for class c_1 we have $p_{s,c_1} = 1$ for each $s \in S$, and for class c_2 we have $p_{s_1,c_2} = a < 1$, and $p_{s_2,c_2} = 0$. We refer to such a task-expert system as a *two types-two experts* system with parameter a.

For this system, if a task of mixed-type z' receives a failure from either of the experts, then its mixed type becomes z'' where $z''_{c_1} = 0$ and $z''_{c_2} = 1$. Thus, it is sufficient to assume that $\mathcal{Z} = \{z', z''\}$ where $z'_c = \frac{1}{2}$ for each $c \in C$, and $z''_c = \mathbf{1}\{c = c_2\}$. Further, it is easy to check that $\psi_{s_1}(z') = (1 - a)/2$, $\psi_{s_1}(z'') = 1 - a$, $\psi_{s_2}(z') = 1/2$, and $\psi_{s_2}(z'') = 1$. We then have the following result (its proof, as that of the other stability results in the article, is established through the identification of suitable Lyapunov functions, and given in the Section 8).

Theorem 3.3. *For a two types-two experts system with parameter a as defined in Definition 3.2, the system under Greedy is stable if and only if $\lambda < 4a/(2 + a)$.*

3.2 Random Policy

Definition 3.4 (Random Policy). A policy is Random if each expert s is assigned a task chosen uniformly at random from the pool of outstanding tasks.

This policy is particularly easy to implement as it does not require knowledge of any system parameter. The following theorem provides a stability conditions for this policy.

Theorem 3.5. *Under Random policy, the system is stable if and only if the following holds:*

$$\lambda < \left(\sum_{c \in C} \frac{\sum_{z \in \mathcal{Z}} z_c \pi_z}{\sum_{s \in S} \mu_s p_{s,c}} \right)^{-1},$$

Following corollary easily results from the above theorem.

Corollary 3.6. *Consider a two types-two experts system with parameter a as defined in Definition 3.2. The system is stable under Random policy if and only if* $\lambda < 4a/(2+a)$.

3.3 Sub-optimality of Greedy and Random

The result in Corollary 3.6 implies that for the two types-two experts system with parameter a considered in Theorem 3.3, Greedy policy performs not better than Random policy. Further, we show in the next section, via Theorem 4.1 and Corollary 4.4, that there exists a policy which achieves optimal stability region which is significantly larger than that achieved by Greedy and Random. In particular, the stability threshold for task arrival rates under optimal policy can be up to 5/4 times (namely, when $a = 1/2$) that under either Greedy or Random.

While it is intuitive that Random may perform poorly as compared to an optimal policy, it is somewhat counter intuitive that Greedy may perform as sub-optimally as Random. The reason for the poor performance of Greedy can be explained as follows. In the two types-two experts system, we have a flexible expert, i.e. an expert for tasks of all pure-types, and a specialized expert, i.e. an expert only for pure-type c_1. Under Greedy policy, all experts prioritize the newly arriving tasks as it optimizes the probability of achieving success in the short term. However, a larger long-term throughput could be achieved if the flexible expert could focus more on the lagging tasks, i.e., the tasks of pure-type c_2.

4 OPTIMAL STABILITY

Main goal of this section is to provide necessary and sufficient conditions for stability of the system. We provide a policy, called backpressure policy, which stabilizes the system when the sufficient conditions are satisfied.

We obtain stability conditions via capacity constraints and flow conservation constraints which capture the flow of tasks from one type to another upon service by an expert. For instance, if $\nu_{s,z}$ represents the flow of tasks of mixed-type z served by expert s, a fraction $1-\psi_s(z)$ of it leaves the system due to success while the rest gets converted into a flow of type $\phi_s(z)$. The total arrival rate of flow of mixed-type z, i.e., $\lambda\pi_z + \sum_{s\in S, z'\in\phi_s^{-1}(z)} \nu_{s,z'}\psi_s(z')$, must match the total service rate, i.e., $\sum_{s\in S}\nu_{s,z}$. Further, the total flow service rate expert s, i.e., $\sum_{z\in\mathcal{Z}}\nu_{s,z}$, must be less than its service capacity μ_s. The following is the main result of this section. For a proof see Section 8.3

Theorem 4.1. *Suppose there exists s such that* $\min_c p_{s,c} > 0$. *If there exist non-negative real numbers* $\nu_{s,z}$, *for* $s \in S$, *and* $z \in \mathcal{Z}$ *and positive real numbers* δ_s, *for* $s \in S$ *such that the following hold:*

$$\forall z \in \mathcal{Z}, \quad \lambda\pi_z + \sum_{s\in S, z'\in\phi_s^{-1}(z)} \nu_{s,z'}\psi_s(z') = \sum_{s\in S}\nu_{s,z}, \tag{3}$$

$$\forall s \in S, \quad \sum_{z\in\mathcal{Z}}\nu_{s,z} + \delta_s \le \mu_s, \tag{4}$$

then there exists a policy under which the system is stable. If there does not exist non-negative real numbers $\nu_{s,z}$, *for* $s \in S$, $z \in \mathcal{Z}$ *and non-negative real numbers* δ_s *for* $s \in S$ *such that the above constraints hold, then the system cannot be stable.*

We use the condition of existence of an expert s such that $\min_c p_{s,c} > 0$ only for a technical reason to simplify our proof. We believe that the result holds even when this is not true.

We now describe the policy which achieves optimal stability. We need some more notation to describe the policy. Consider a set $\mathcal{Y} \subset \mathcal{Z}$. Let $X(t)$ be the number of tasks in the system at time t which have been of type $z \in \mathcal{Z}\backslash\mathcal{Y}$. For $z \in \mathcal{Y}$, let $\tilde{X}_z(t)$ and $\tilde{N}_z(t)$ be the tasks of mixed-type z which have and have not had mixed-type in $\mathcal{Z}\backslash\mathcal{Y}$. Also, for convenience, for each $z \in \mathcal{Z}\backslash\mathcal{Y}$, let $\tilde{X}_z$ be the tasks of mixed-type z, i.e., $N_z = \tilde{X}_z$ for each $z \in \mathcal{Z}\backslash\mathcal{Y}$. Thus, we have $X = \sum_z \tilde{X}_z$. Consider the following policy.

Definition 4.2 (Backpressure($\mathcal{Y}$) policy). For a given $\mathcal{Y}$, let X, and $\{\tilde{N}_z\}_{z\in\mathcal{Y}}$ be as defined above. For each $s \in S, z \in \mathcal{Y}$ let

$$w_{s,z}(\tilde{N}, X) = \begin{cases} \tilde{N}_z - \psi_s(z)\tilde{N}_{\phi_s(z)}, & \text{if } \phi_s(z) \in \mathcal{Y} \\ \tilde{N}_z - \psi_s(z)X, & \text{if } \phi_s(z) \in \mathcal{Z}\backslash\mathcal{Y} \end{cases}.$$

Define

$$B_s(\tilde{N}, X) = \arg \max_{z' \in \mathcal{Y}: \tilde{N}_{z'} > 0} w_{s,z}(\tilde{N}, X).$$

If

$$\sum_s \mu_s \max_{z \in \mathcal{Y}: \tilde{N}_z > 0} w_{s,z}(\tilde{N}, X) \geq X \min_{c \in C} \sum_s \mu_s p_{s,c}$$

then each expert chooses a task in $\tilde{N}_z$ where $z \in B_s(\tilde{N}, X) \subset \mathcal{Y}$ with ties broken arbitrarily. Else, each expert serves a task in X chosen uniformly at random.

The following theorem follows from the proof of Theorem 4.1.

THEOREM 4.3. *Suppose there exists s such that* $\min_c p_{s,c} > 0$. *If the sufficient conditions for stability as given by Theorem 4.1 are satisfied, then there exists a finite subset $\mathcal{Y}$ of $\mathcal{Z}$ such that the policy Backpressure($\mathcal{Y}$) stabilizes the system.*

Unlike backpressure policy proposed in [33] under a different setting, which was agnostic to system arrival rates, a set $\mathcal{Y}$ such that Backpressure($\mathcal{Y}$) policy stabilizes the system may depend on the value of λ. We now use Theorem 4.1 to provide sufficient conditions for stability for a scenario considered in Section 3.1. For proof of the corollary below, see Section 8.5.

COROLLARY 4.4. *For a two types-two experts system with parameter a as defined in Definition 3.2, the Backpressure($\mathcal{Z}$) policy stabilizes the system if* $\lambda < \min\{3a/(a+1), 2a\}$. *Further, the system is unstable if the strict inequality is reversed.*

Compare this with the stability region achieved by greedy and random policies, as given by Theorem 3.3 and Corollary 3.6, respectively. As we argued in Section 3.3, the backpressure($\mathcal{Y}$) policy may significantly outperform greedy and random policies.

5 EXPERIMENTAL RESULTS

In this section, we present our empirical results obtained by using data from Math.Stack-Exchange Q&A platform. In this platform, users post tagged questions that are answered by other users. The asker may select one of the submitted answers as the best answer for the given question, which is made public information in the platform. We will use this data to estimate the success probabilities of experts in answering questions, and use this parameters in simulations to compare the throughputs that can be achieved by our two baseline policies and the stability-optimal backpressure policy. This will show that a substantially larger throughput can be achieved by the stability-optimal backpressure policy relative to the baseline policies.

Dataset. The dataset consists of around 800, 000 questions. It was retrieved on February 2nd, 2017. The top 11 most common tags are given in Table 1 in decreasing order of popularity. Among these tags, the most common is *calculus* which covers 61, 184 questions, and the least common is *complex analysis* which covers 22, 813 questions. In our analysis, we used only questions that are tagged with at least one of the 11 most popular tags, which amounts to a total of 381, 239 questions.

Table 1. Skills of experts estimated by using data from the Math.Stack-Exchange Q&A platform.

	Clusters									
Tag	1	2	3	4	5	6	7	8	9	10
calculus	.32	**.39**	.30	**.35**	**.37**	**.47**	.28	.16	.26	**.41**
real-analysis	.17	**.41**	.25	.32	.23	**.49**	**.40**	.10	.10	**.44**
linear-algebra	**.46**	.29	.05	**.36**	.14	**.48**	.26	.31	.07	**.43**
probability	.07	**.49**	.02	.33	.02	**.50**	.06	.02	**.46**	.04
abstract-algebra	.02	.05	.03	.32	.02	**.38**	.23	**.50**	.01	.27
integration	.09	**.43**	.05	.19	**.44**	**.45**	.03	.01	.06	**.37**
sequences-and-series	.05	.32	.16	.31	.20	**.45**	.09	.04	.06	.33
general-topology	.02	.10	.03	.16	.02	**.43**	**.50**	.07	.02	.31
combinatorics	.03	.14	.06	**.43**	.04	**.37**	.02	.06	.19	.05
matrices	.27	.15	.02	.31	.02	**.44**	.06	.11	.02	.34
complex-analysis	.02	.19	.08	.16	.14	**.50**	.09	.05	.01	**.44**
Size	165	188	313	200	179	183	231	187	178	176

Estimated skill sets. The success probabilities of answering questions are estimated as follows. For a given user-tag pair, the success probability is estimated by the empirical frequency of the accepted answers by this user for questions of given tag, conditional on that the user had at least 5 accepted answers for questions of the given tag, and otherwise we estimate the success probability is set to be equal to zero. These success probabilities are estimated for 2,000 users with the most accepted answers. Among these users, the user with the most accepted answers had 4,665 accepted answers, and the user with the least number of accepted answers had 13 accepted answers. 712 users had more than 50 answers accepted. In order to form clusters of users with similar success probabilities for different tags, we ran the k-means clustering algorithm.

The estimated success probabilities are shown in Table 1. The columns correspond to different centroids of the clusters and give average success probabilities for different tags. In the bottom row, we give the sizes of the corresponding clusters. For instance, the 165 persons in cluster 1 have on average 32% of their *calculus*, and 46% of their *linear algebra* answers accepted.

There is a pronounced separation of users who answer questions with respect to their expertise into those who are good at answering questions of any topic and those who are specialized in particular topics. In Table 1, we highlighted the success probabilities of value at least 35% from which we observe that (a) there is a cluster, namely cluster 6, which contains 183 very active, experienced experts doing well on all tags; (b) there is a cluster, namely cluster 3, which contains a bulk of 313 users with low performance over all tags; (c) there are clusters, namely clusters 1, 7, 8 and 9, which contain users specialized in a single topic; and, finally, (d) the users in remaining clusters, namely clusters 2, 4, 5 and 10 do well on a few related topics.

There is also a prevalence of questions with different combinations of tags, that is, mixed types. We kept only those combinations of tags that occur for at least 1% of the total number of questions. This results in 16 tag combinations among which 11 are singletons and 5 are combinations of 2 tags. For instance, 11% of the questions are tagged only with *calculus* and 4% of the questions are tagged with both *linear-algebra* and *matrices*. We observed that roughly 19% of the questions are tagged with multiple tags, showing the relevance of our model.

Simulation setup. We regard 10 clusters of users as the set of *experts* in our system and consider the 11 tags as *pure* types; the questions are then either of pure or *mixed* type. We define the set of task types as the set containing all pure types plus the 5 most frequently seen mixed types. For a mixed type we assume that the priors are uniformly distributed (i.e., a question with tags *calculus* and *integration* is with 1/2 chance of pure type *calculus*). Each arrival task is one of 16 types plus all vectors that can be obtained by applying a finite composition of functions in $\{\phi_s(\cdot)\}_{s\in S}$ on the arriving types. We defined the arrival frequencies of task types in our simulations by normalizing the occurrence of the 16 types. We assumed the experts to have unit service rates. We make this approximation as we do not have the information about times at which experts begin to respond a question. We examined the system for increasing values of task arrival rates. We simulate our CTMC via a discrete event simulator. It sequentially computes changes of the state of the system as determined by the occurrence of some event such as a task arrival or a response from an expert. All our experiments were for a simulation run that consists of 3.5 million events.

For the backpressure policy we define the set $\mathcal{Y}$ to consist of all 11 pure types, the 5 most frequently seen mixed types upon arrival as described above, and the mixed types that result from addressing the above mixed types by an expert exactly once. Note that pure types can be attempted multiple times without changing its type. We thus have $|\mathcal{Y}| = 16 + 5 \cdot 10 = 66$. Our choice of $\mathcal{Y}$ is a result of a compromise between performance and complexity. Choosing a larger set of $\mathcal{Y}$ may increase the stability region by a small fraction, but may significantly increase the complexity of the Backpressure policy.

Performance comparison of different policies. We evaluated the time-average number of tasks in the system for different task arrival rates for the three policies under our consideration. These results are shown in Figure 1. We observe that the task arrival rates at which random, greedy, and backpressure become unstable are, 2.2, 3.80 and 4.10, respectively. Thus, random policy performs much worse than any other policy, and the backpressure policy achieves throughput improvement of about 8% over the greedy policy.

We further examined the evolution of the number of tasks in the system waiting to be served over time for greedy and backpressure policy for respective task arrival rates, 3.78 and 3.83 (Figure 2 left) and respective task arrival rates 3.83 and 4.08 (Figure 2 right). We observe that for the two comparison cases the backpressure policy tends to result in smaller number of tasks waiting to be served than the greedy policy, even when operating under a larger task arrival rate. By experimentations, we observed that under the backpressure policy the system becomes unstable at an arrival rate of about 4.1. We expect that extending the definition of $\mathcal{Y}$ would allow to achieve even higher throughputs.

6 RELATED WORK

The problem studied in this paper is broadly related to that of *multi-arm bandits*, e.g., see [1, 4, 12, 20] and citations therein, in the sense of optimizing the trade-off between exploration, to learn job types, and exploitation, to optimize task performance. It also has some relation with *collaborative filtering systems* such as those studied in [18, 19, 32], which can be interpreted as expert-task systems where success probabilities admit a low-rank matrix structure. Unlike our work, there good matches are inferred from observed assignments of tasks to experts, which are according to a given statistical model, and there are no resources constraints imposed on the experts.

A related line of work is that on *stochastic online matching*, e.g., [14, 25, 26]. The stochastic online matching can be interpreted as a task-expert system where each expert is associated with a budget constraint that allows to solve at most one task. The goal is to maximize the expected number of successful assignments over a given number of task arrivals. The competitive ratios of different assignment policies have been established under certain restrictions on the success probabilities, such as assuming that each takes either a common positive value or zero value, or that each takes an arbitrary but small enough value. Unlike our work where the task types are uncertain, uncertainty in these models come from the arbitrariness of the future task arrivals and the monotonically decreasing available resource budgets.

In [5], the authors considered a task-expert system where task types are of two difficulty levels (hard or easy) and expert skills are of two levels (senior or junior). Seniors may serve any task, but juniors may only serve easy

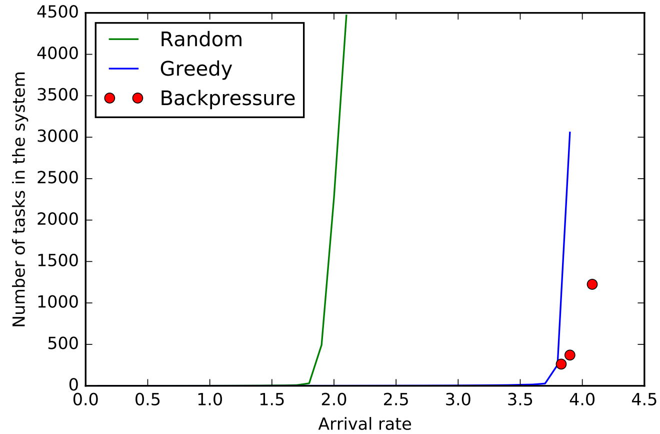

Fig. 1. Time-average number of tasks in the system versus the task arrival rate for three different policies: random, greedy and backpressure. The system under random policy becomes unstable at task arrival rate of about 2.2, under greedy policy at about 3.8 and under backpressure policy of about 4.1 (not shown in the figure). Note that the backpressure policy still allows for the system to be stable at the task arrival rate of about 4.08 (the right-most red dot in the figure).

tasks. The type of each task is assumed to be unknown upon arrival. It is shown that there exists an optimal policy where all tasks are first routed to junior experts and then progressively moved to senior ones. We allow for much more generality with respect to the heterogeneity of skills of experts. In their model, a task upon service can only become progressively harder, which amounts to a feed-forward system, unlike our model.

The work in [16] considers a model where the expert resources are constrained and shared by different tasks of uncertain types. They consider a setting where each task can be divided into a large number of subtasks of the same type, vanishingly small amount of which could be used to accurately learn the task's type, and the rest can be served optimally. This decouples the learning of each task's type from the performance in execution and expert utilization, and is thus different from our work.

Another related literature is that of *constrained queueing systems*, where arriving tasks are to be served by heterogeneous servers subject to resource constraints, e.g., [2, 7, 8, 11, 15, 21, 27, 31, 35]. The goal is to efficiently utilize server resources while providing good performance in servicing tasks, e.g., optimizing task delays. Our matching policy is of a flavor similar to the stability-optimal backpressure policy first proposed in [33]. The setting close to ours is the one studied in [31] for *routing queries in peer-to-peer networks*. Here, the types of the queries are known but the locations of nodes where the queries may by successfully resolved are uncertain. More technically, we associate queues with each prior distribution which may be infinite in number. This makes the stability analysis much more challenging. Another related work is that on *scheduling flexible servers* [22], which allows for tasks of different types and servers of different skills. It has been established that a greedy, so called max-weight policy, minimizes a strictly convex cost function of task delays in a heavy traffic regime. In [13] this strictness requirement is removed for a class of systems with homogeneous server speeds. The main difference from our work is that all these works assume that task types are known.

Finally, we contrast our work to the matching problem studied in the context of crowdsourcing systems such as [10, 17, 30, 36]. These works are concerned with classification tasks with unknown ground truths. This is unlike our work, where we consider tasks such that upon each attempt of an expert solving a task, we observe whether or not the task has been successfully solved. Another difference is that these works typically consider a static

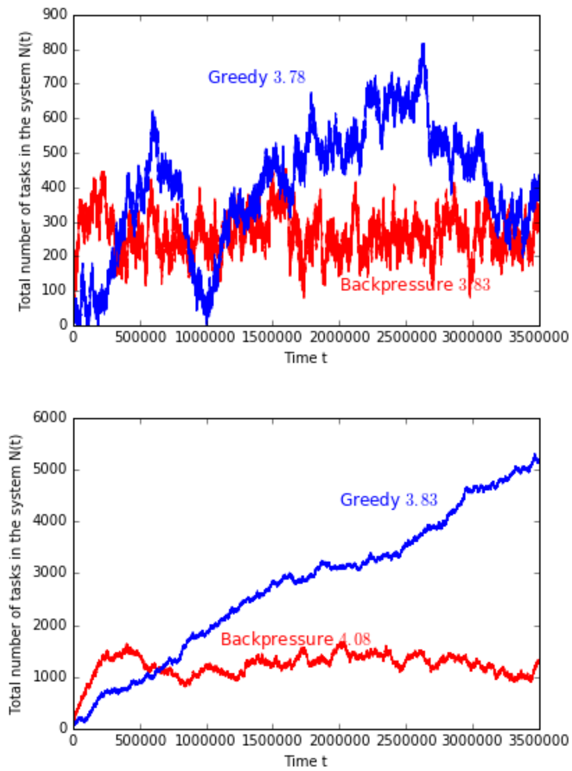

Fig. 2. Total number of tasks in the system over time for the greedy and backpressure policy. The blue curves correspond to greedy policy and red curves correspond to backpressure policy. The task arrival rates are as indicated in the figures. (Left) both policies provide stability but backpressure policy has overall better performance. (Right) greed policy fails to provide stability, while backpressure policy provides stability.

model where a task assignment to a set of experts is made all at once. In [24] labeling tasks arrive dynamically and their exit is tied to the expert allocation decisions, in that a task leaves once the probability of error in the label estimate falls below a threshold.

7 CONCLUSION

We studied matching of tasks and experts in a system with uncertain task types. We established a complete characterization of the stability region of the system, i.e. the set of task arrival rates that can be supported by a matching policy such that the expected number of tasks waiting to be served is finite. We showed that any task arrival rate in the stability region can be supported by a back-pressure matching policy. We also compared with two baseline matching polices, and identified instances under which there is a substantial gap between the maximum task arrival rates that can be supported by these policies and that of the optimum back-pressure matching policy.

There are several interesting directions for future research. First, for the case when task types are unknown, it is of interest to consider matching policies that optimise different kinds of performance objectives, such as, for example, minimizing the long-run average of a function of task waiting times. Second, much remains to be said about matching policies for the case when both task types and the skills of experts are unknown.

8 PROOFS

8.1 Proof of Theorem 3.3

We first show that if $\lambda < 4a/(2+a)$ then the system is stable. For each t let $t + \tau(t)$ be the time at which the first event (arrival or completion of a response) occurs after time t. Let $\tau_n = E[\tau(t)|N(t) = n]$, i.e., given that $N(t) = n$ at time t, τ_n is the expected time at which the first event occurs after time t. For example, for $n = 0$ we have $\tau_n = 1/\lambda$.

A common approach to show system stability is to use Lyapunov-Foster theorem, see e.g., Proposition I.5.3 on page 21 in [3]. The idea is to construct a function $L(\cdot)$ such that $L(n)$ tends to infinity as $|n| \to \infty$ and that has a strictly negative 'drift' for all but finite values of n, i.e., there exists a constant $\epsilon > 0$ such that

$$E\left[L(N(t+\tau(t))) - L(N(t))\middle|N(t) = n\right] \le -\epsilon\tau_n,$$

for all but finite values of n. Intuitively, negative drift condition implies that as $L(N(t))$ becomes large (i.e., as $N(t)$ becomes large) the system dynamics is such that $L(N(t))$ tends to decrease in expectation. This prevents the $L(t)$ from blowing up to ∞ as t increases and thus keeps the system stable. Roughly, the Lyapunov-Foster theorem states that the negative drift condition is indeed sufficient to ensure that that system is positive recurrent and thus stable. We will use a variant of Lyapunov-Foster theorem, provided below, which follows from Theorem 8.13 in [28].

Theorem 8.1. *Consider an irreducible CTMC $N(t)$ that takes values in a countable state-space. Let $\tau(t)$ and τ_n be as defined above. If there exists a function $L(\cdot)$, and constants $K > 0$ and $\epsilon > 0$ such that for $L(n) > K$ we have*

$$E\left[L(N(t+\tau(t))) - L(N(t))\middle|N(t) = n\right] \le -\epsilon\tau_n,$$

and if $\{n : L(n) \le K\}$ is finite, then $N(t)$ is positive recurrent.

Now suppose that $\lambda < 4a/(2+a)$. Then, it can be checked that $\frac{2-a}{2(2-\lambda)}\lambda < a$. Thus, there exists $\delta > 0$ such $\frac{2-a}{2(2-\lambda)}\lambda = (1+\delta)a$. Now, consider the following Lyapunov function: for each n, we have

$$\frac{1}{\tau_n}L(n) = (1+\delta)\frac{2-a}{2(2-\lambda)}n_{z'} + n_{z''},$$

where δ is a constant obtained as above.

Consider states n such that $n_{z'} > 0$. For these states, we obtain

$$\begin{aligned}\frac{1}{\tau_n}L(n) &= (1+\delta)\frac{2-a}{2(2-\lambda)}\left(\lambda - \mu_{s_1} - \mu_{s_2}\right) + \left(\mu_{s_1}\psi_{s_1}(z') + \mu_{s_2}\psi_{s_2}(z')\right)\\ &= (1+\delta)\frac{2-a}{2(2-\lambda)}(\lambda-2) + \frac{1-a}{2} + \frac{1}{2} = -\delta\frac{2-a}{2} < 0.\end{aligned}$$

Now, consider states n such that $n_{z'} = 0$ and $n_{z''} > 0$. For these states

$$\frac{1}{\tau_n}L(n) = \delta\frac{2-a}{2(2-\lambda)}\lambda - \mu_s s_1 a = -\delta a < 0.$$

Thus, the conditions of Theorem 8.1 are satisfied with $K = L((1,1))/\tau_{(1,1)}$ and $\epsilon = \min(\delta a, \delta(2-a)/2)$. Hence, $N(t)$ is positive recurrent if $\lambda < 4a/(2+a)$.

We now show the only if part. Suppose that $\lambda \ge 4a/(2+a)$. Then, the δ used in the above argument is greater than or equal to 0. Thus, drift is non-negative for all but finite values of n. Further, since $L(\cdot)$ is bounded, the maximum change in $L(\cdot)$ upon an arrival or a departure is also bounded, using Proposition I.5.4 on page 22 in [3], we establish the only if part. □

8.2 Proof of Theorem 3.5

Note that the system under random policy is equivalent to the one where pure-type of a task is revealed upon arrival, i.e., there is no uncertainty in task types. This is true since the random policy does not use the information of type (pure or mixed). We thus let that pure-type is indeed revealed upon arrival. Let $X_c(t)$ be the number of tasks in the system of pure-type c. Let $X(t) = \{X_c(t)\}_c$. For each $c \in C$, the arrival rate into queue $X_c(t)$ is equal to

$$\lambda_c \triangleq \sum_{z \in Z} \lambda z_c \pi_z.$$

We first show the if part of the result. Suppose that we have $\frac{\sum_{c \in C} \lambda_c}{\sum_{s \in S} \mu_s p_{s,c}} < 1$. We use the fluid limit approach developed in [9, 23, 29]. Roughly, given initial condition $X(0) = x$, the fluid trajectories of the state process $X(t)$ can be obtained by scaling initial conditions, speeding time, and then studying the rescaled process; i.e., letting $\lim_{k \to \infty} \frac{1}{k} X(0) = x$, and studying $\lim_{k \to \infty} \frac{1}{k} X(kt)$.

Using arguments similar to those used in [23], the fluid limits for the number of tasks in each class can be shown to satisfy the following at almost all times t: for each $c \in C$ and $X_c > 0$ we have

$$\frac{d}{dt} X_c = \lambda_c - \sum_{s \in S} \mu_s p_{s,c} \frac{X_c}{\sum_{c'} X_{c'}}. \tag{5}$$

Define a function L on $\mathbb{R}^C$ as

$$L(X) = \sum_c X_c \log \left(\frac{X_c}{\gamma_c \sum_{c'} X_{c'}} \right), \tag{6}$$

where $\gamma_c \triangleq \frac{\lambda_c}{\sum_{s \in S} \mu_s p_{s,c}}$.

Further, by following the arguments similar to [23], if we have that $L(X) \to \infty$ and $\frac{d}{dt} L(X) \to -\infty$ as $|X| \to \infty$ under fluid limits then the stability of the original system follows. We show below that both these limits hold.

Using (5) and (6), we obtain

$$\begin{aligned}
\frac{d}{dt} L(X) &= \sum_c \left(\frac{d}{dt} X_c \right) \log \left(\frac{X_c}{\gamma_c \sum_{c'} X_{c'}} \right), && (7)\\
&= \sum_c \left(\lambda_c - \sum_{s \in S} \mu_s p_{s,c} \frac{X_c}{\sum_{c'} X_{c'}} \right) \left(\log \frac{X_c}{\sum_{c'} X_{c'}} - \log \gamma_c \right), && (8)\\
&= \sum_c \left(\sum_s \mu_s p_{s,c} \right) \left(\frac{\lambda_c}{\sum_{s \in S} \mu_s p_{s,c}} - \frac{X_c}{\sum_{c'} X_{c'}} \right) \left(\log \frac{X_c}{\sum_{c'} X_{c'}} - \log \gamma_c \right) && (9)
\end{aligned}$$

which converges to $-\infty$ as $|X|$ grows large.

Let $\theta = 1/\sum_c \gamma_c$ and $\hat{\gamma}_c = \gamma_c/\theta$ for each $c \in C$. Since $\sum_c \gamma_c < 1$, we have $\theta > 1$. Let $D(p||q)$ be the Kullback-Leibler divergence between two Bernoulli distributions with parameters p and q, i.e., $D(p||q) = p \log(\frac{p}{q}) + (1 - p) \log(\frac{1-p}{1-q})$. Now, we can write

$$L(X) = \sum_c X_c \log\left(\frac{\theta X_c}{\hat{\gamma}_c \sum_{c'} X_{c'}}\right) \tag{10}$$

$$= \sum_c X_c \log\theta + \sum_c X_c \log\left(\frac{X_c}{\sum_{c'} X_{c'}}.\frac{1}{\hat{\gamma}_c}\right) \tag{11}$$

$$= \sum_c X_c \log\theta + \left(\sum_c X_c\right) D\left(\left\{\frac{X_c}{\sum_{c'} X_{c'}}\right\}_{c\in C} \middle\| \{\hat{\gamma}_c\}_{c\in C}\right) \tag{12}$$

which converges to ∞ as $|X|$ grows large.

Hence, the if part of the result follows. The same line of argument can also be used to show that if $\frac{\sum_{c\in C}\lambda_c}{\sum_{s\in S}\mu_s p_{s,c}} \geq 1$ the drift is non-negative for all but finite number of states. Further, since $L(X)$ is bounded, the maximum change in $L(X)$ upon an arrival or a departure is also bounded, using Proposition I.5.4 on page 22 in [3], we get the only if part. □

8.3 Proof of Theorem 4.1

We first show stability under sufficient conditions. In networked systems, e.g. see [11, 33], a standard approach towards proving stability of a backpressure type policy is to design a 'static' policy using flow variables $\{\nu_{sz}\}_{s,z}$ and the slacks $\{\delta_s\}_s$ which provides a fixed service rate to each queue N_z such that its drift is sufficiently negative for each. However, in our setup the total number of queues $\{N_z\}_{z\in\mathcal{Z}}$ could be countable, while the total available slack is finite. Thus, it is not possible to design a static policy such that the drift in each individual queue is bounded from above by a negative constant. This is unlike any finite-server queueing system considered in the previous literature.

We thus take a different approach, which can be explained roughly as follows. Since the total exogenous arrival rate λ, and the total endogenous arrival rate, i.e. arrival into a queue due to failure at another queue, are both finite (they are bounded from above by $\sum_s \mu_s$), there exists a finite set $\mathcal{Y} \subset \mathcal{Z}$ such that the total arrival rate into $\mathcal{Z}\backslash\mathcal{Y}$ is less than $\min_{c\in C}\sum_{s\in S}\frac{\delta_s}{4}p_{s,c}$. Each task which enters a queue N_z where $z \in \mathcal{Z}\backslash\mathcal{Y}$ is instead sent to a virtual queue X, and stays there until there is a success. If X is 'large' compared to the other queues then all the servers focus on X. The finite number of remaining queues are operated via a backpressure policy which accounts for the 'expected backlog' seen in these queues.

More formally, consider $\{\nu_{s,z}\}_{s,z}$ and positive constants $\{\delta_s\}_s$ as postulated in the theorem. Without loss of generality, assume that there exists a constant $0 < \epsilon < 1$ such that $\delta_s = \epsilon\mu_s$ for each $s \in S$. Let $\mathcal{Y}$ be a finite subset of $\mathcal{Z}$ such that

$$\sum_{z\in\mathcal{Z}\backslash\mathcal{Y}}\left(\lambda\pi_z + \sum_{s\in S}\sum_{z'\in\phi_s^{-1}(z)\cap\mathcal{Y}} \nu_{s,z'}\psi_s(z')\right) \leq \min_{c\in C}\sum_{s\in S}\frac{\delta_s}{4}p_{s,c}.$$

Since $\lambda + \sum_{s\in S, z\in\mathcal{Z}} \nu_{s,z} \leq 2\sum_s \mu_s$, such a $\mathcal{Y}$ exists.

Let X be the number of tasks in the system which are or have been in past of type $z \in \mathcal{Z}\backslash\mathcal{Y}$. Once a task enters queue X it does not leave it until success. There may be tasks in it with mixed-type in $\mathcal{Y}$. Note, our policy will depend on X and thus $\{z(s,t)\}_s$ will not be $N(t)$ measurable. In turn, $N(t)$ will not be a CTMC. For $z \in \mathcal{Y}$, let $\tilde{X}_z$ and $\tilde{N}_z$ be the tasks of mixed-type z which have and have not had mixed-type in $\mathcal{Z}\backslash\mathcal{Y}$. Also, for convenience for each $z \in \mathcal{Z}\backslash\mathcal{Y}$, let $\tilde{X}_z$ be the tasks of mixed-type z, i.e., $N_z = \tilde{X}_z$ for each $z \in \mathcal{Z}\backslash\mathcal{Y}$. We now formally define $\sigma\left(\{\tilde{X}_z\}_{z\in\mathcal{Z}}, \{\tilde{N}_z\}_{z\in\mathcal{Y}}\right)$-measurable backpressure policy. Thus, $\left(\{\tilde{N}_z\}_{z\in\mathcal{Y}}, \{\tilde{X}_z\}_{z\in\mathcal{Z}}\right)$ is a CTMC.

We now show stability of the system under this policy for Backpressure($\mathcal{Y}$) as given in Definition 4.2. Below we will assume that the ties in selecting z from $B_s(\tilde{N}, X)$ are broken uniformly at random for simplicity of exposition. The proof can be easily extend to any other tie breaking approach. Consider the following Lyapunov function.

$$L(\tilde{N}, \tilde{X}) = \sum_{z \in \mathcal{Y}} \tilde{N}_z^2 + \left(\sum_{z \in \mathcal{Z}} X_z \right)^2 = \sum_{z \in \mathcal{Y}} \tilde{N}_z^2 + X^2.$$

For each t, let $t + \tau(t)$ be the time at which the first event (arrival or completion of a response) occurs after time t. Clearly, $\tau(t)$ is a stopping time. Further, let $\tau_{\tilde{n},\tilde{x}}(t) = E[\tau(t) | (\tilde{N}, \tilde{X}) = (\tilde{n}, \tilde{x})]$.

Let

$$D(\tilde{n}, \tilde{x}) \triangleq \frac{1}{\tau_{\tilde{n},\tilde{x}}} E \left[L(\tilde{N}(t+\tau), \tilde{X}(t+\tau)) - L(\tilde{N}(t), \tilde{X}(t)) \middle| \tilde{N}(t) = \tilde{n}, \tilde{X}(t) = \tilde{x} \right].$$

$D(\tilde{n}, \tilde{x})$ is called drift in state n. We would like to show that there exists a positive integer K and positive constant ϵ such that

$$D(\tilde{n}, \tilde{x}) \leq -\epsilon \quad \forall (\tilde{n}, \tilde{x}) \text{ s.t. } \max(|\tilde{n}|_\infty, x) \geq K.$$

Let for each $s \in S$ and $z \in \mathcal{Y}$ let

$$\nu^*_{s,z} = \mathbf{1}\left\{ x \min_{c \in C} \sum_s \mu_s p_{s,c} > \sum_s \mu_s \max_{z \in \mathcal{Y}: \tilde{n}_z > 0} w_{s,z}(\tilde{n}, x) \right\} \mathbf{1}\{z \in B_s(n)\} \frac{1}{|B_s(n)|}.$$

Then, one can check that

$$\frac{1}{\tau_{\tilde{n},\tilde{x}}} E[\tilde{N}_z(t+\tau)^2 - \tilde{N}_z(t)^2 \big| \tilde{N}(t) = \tilde{n}, \tilde{X}(t) = \tilde{x}] = (2\tilde{n}_z + 1) \left(\lambda \pi_z + \sum_{s \in S} \sum_{z' \in \phi_s^{-1}(z) \cap \mathcal{Y}} \nu^*_{sz'} \psi_s(z') \right) + (-2\tilde{n}_z + 1) \sum_s \nu^*_{s,z}.$$

Further, let

$$\nu^* = \mathbf{1}\left\{ x \min_{c \in C} \sum_s \mu_s p_{s,c} > \sum_s \mu_s \max_{z \in \mathcal{Y}: \tilde{n}_z > 0} w_{s,z}(\tilde{n}, x) \right\}.$$

Then, we have that

$$\begin{aligned} &\frac{1}{\tau_{\tilde{n},\tilde{x}}} E[X(t+\tau)^2 - X(t)^2 \big| \tilde{N}(t) = \tilde{n}, \tilde{X}(t) = \tilde{x}] \\ &\qquad \leq (2x+1) \sum_{z \in \mathcal{Z} \setminus \mathcal{Y}} \left(\lambda \pi_z + \sum_{s \in S} \sum_{z' \in \phi_s^{-1}(z) \cap \mathcal{Y}} \nu^*_{s,z'} \psi_s(z') \right) + (-2x+1) \nu^* \min_c \sum_s \mu_s p_{s,c} \end{aligned}$$

Thus, we get

$$\begin{aligned} D(\tilde{n}, \tilde{x}) \leq &\sum_{z \in \mathcal{Y}} (2\tilde{n}_z + 1) \left(\lambda \pi_z + \sum_{s \in S} \sum_{z' \in \phi_s^{-1}(z) \cap \mathcal{Y}} \nu^*_{s,z'} \psi_s(z') \right) + (-2\tilde{n}_z + 1) \sum_s \mu_s \nu^*_{s,z} \\ &+ (2x+1) \sum_{z \in \mathcal{Z} \setminus \mathcal{Y}} \left(\lambda \pi_z + \sum_{s \in S} \sum_{z' \in \phi_s^{-1}(z) \cap \mathcal{Y}} \nu^*_{s,z'} \psi_s(z') \right) + (-2x+1) \nu^* \min_c \sum_s \mu_s p_{s,c}. \end{aligned}$$

Upon arranging terms, we obtain

$$
\begin{aligned}
D(\tilde{n},\tilde{x}) \le \sum_{z\in\mathcal{Y}} 2\tilde{n}_z \left(\lambda\pi_z + \sum_{s\in S}\sum_{z'\in\phi_s^{-1}(z)\cap\mathcal{Y}} v^*_{sz'}\psi_s(z') - \sum_s v^*_{s,z}\right) \\
+ 2x\left(\sum_{z\in\mathcal{Z}\setminus\mathcal{Y}}\left(\lambda\pi_z + \sum_{s\in S}\sum_{z'\in\phi_s^{-1}(z)\cap\mathcal{Y}} v^*_{sz'}\psi_s(z')\right) - v^*\min_c\sum_s \mu_s p_{s,c}\right) \\
+\left(\lambda + \sum_{z\mathcal{Z}}\sum_{s\in S}\sum_{z'\in\phi_s^{-1}(z)\cap\mathcal{Y}} v^*_{s,z'}\psi_s(z') + \sum_{z\in\mathcal{Y}}\sum_s v^*_{s,z} + v^*\min_c\sum_s \mu_s p_{s,c}\right)
\end{aligned}
$$

The last of the above three terms can be bounded by a constant, say $\alpha_1 = 10\sum_s \mu_s$. For each $s \in S$ and $z \in \mathcal{Y}$ let $\hat{v}^*_{s,z} = (\mu_s - 3\delta_s/4)v^*_{s,z}$ and $\tilde{v}^*_{sz} = (\delta_s/4)v^*_{s,z}$. Further, let $\hat{v}^* = \min_c\sum_s(\mu_s - 3\delta_s/4)p_{s,c}v^*$ and $\tilde{v}^* = \min_c\sum_s(\delta_s/4)p_{sc}v^*$. Then,

$$
\begin{aligned}
D(\tilde{n},\tilde{x}) \le \sum_{z\in\mathcal{Y}} 2\tilde{n}_z \left(\lambda\pi_z + \sum_{s\in S}\sum_{z'\in\phi_s^{-1}(z)\cap\mathcal{Y}} \hat{v}^*_{sz'}\psi_s(z') - \sum_s \hat{v}^*_{s,z}\right) \\
+ 2x\left(\sum_{z\in\mathcal{Z}\setminus\mathcal{Y}}\left(\lambda\pi_z + \sum_{s\in S}\sum_{z'\in\phi_s^{-1}(z)\cap\mathcal{Y}} \hat{v}^*_{s,z'}\psi_s(z')\right) - \hat{v}^*\right) + \alpha_1 \\
+\sum_{z\in\mathcal{Y}} 2\tilde{n}_z\left(\sum_{s\in S}\sum_{z'\in\phi_s^{-1}(z)\cap\mathcal{Y}} \tilde{v}^*_{s,z}\psi_s(z') - \sum_s \tilde{v}^*_{sz}\right) + 2x\left(\sum_{s\in S}\sum_{z'\in\phi_s^{-1}(z)\cap\mathcal{Y}} \tilde{v}^*_{s,z'}\psi_s(z') - \tilde{v}^*\right)
\end{aligned}
$$

Consider the following lemma. Its proof is given in Section 8.4.

Lemma 8.2. *Recall the $\{v_{s,z}\}_{s,z}$ as postulated by the theorem. For $\Theta = \{\theta_{s,z}\}_{s\in S, z\in\mathcal{Y}} \cup \theta$, where θ and $\theta_{s,z}$ for each s, z are reals, let*

$$
f(\Theta) = \sum_{z\in\mathcal{Y}} 2\tilde{n}_z\left(\lambda\pi_z + \sum_{s\in S}\sum_{z'\in\phi_s^{-1}(z)\cap\mathcal{Y}} \theta_{sz'}\psi_s(z') - \sum_s \theta_{s,z}\right) + 2x\left(\sum_{z\in\mathcal{Z}\setminus\mathcal{Y}}\left(\lambda\pi_z + \sum_{s\in S}\sum_{z'\in\phi_s^{-1}(z)\cap\mathcal{Y}} \theta_{sz'}\psi_s(z')\right) - \theta\right).
$$

Then,

$$
f\left(\{\hat{v}^*_{s,z}\}_{s\in S,z\in\mathcal{Y}} \cup \hat{v}^*\right) \le f\left(\{v_{s,z}\}_{s\in S,z\in\mathcal{Y}} \cup \{\min_s \delta_s/4\}\right).
$$

From definition of $v_{s,z}$, we get that the first term in $f(\{v_{s,z}\}_{s\in S,z\in\mathcal{Y}} \cup \{\min_s \delta_s/4\})$ is equal to 0 and that the second term is less than or equal to 0.

Thus, we obtain

$$
D(\tilde{n},\tilde{x}) \le \alpha_1 + \sum_{z\in\mathcal{Y}} 2\tilde{n}_z\left(\sum_{s\in S}\sum_{z'\in\phi_s^{-1}(z)\cap\mathcal{Y}} \tilde{v}^*_{s,z}\psi_s(z') - \sum_s \tilde{v}^*_{s,z}\right) + 2x\left(\sum_{s\in S}\sum_{z'\in\phi_s^{-1}(z)\cap\mathcal{Y}} \tilde{v}^*_{s,z'}\psi_s(z') - \tilde{v}^*\right).
$$

Rearranging, we get

$$
D(\tilde{n},\tilde{x}) \le \alpha_1 - 2\sum_{s\in S}\sum_{z\in\mathcal{Y}:\phi_s(z)\in\mathcal{Y}} \tilde{v}^*_{sz}(\tilde{n}_z - \psi_s(z)\tilde{n}_{\phi_s(z)}) - 2\sum_{s\in S}\sum_{z\in\mathcal{Y}:\phi_s(z)\in\mathcal{Z}\setminus\mathcal{Y}} \tilde{v}^*_{s,z}(\tilde{n}_z - \psi_s(z)X) - 2x\tilde{v}^*.
$$

Fix $\epsilon > 0$. We now show that there exist a positive integer K such that if $x > K$ or if $|\tilde{n}|_\infty > K$ then $D(\tilde{n},\tilde{x}) \le -\epsilon$. Upon rearranging terms, we obtain

$$
\begin{aligned}
D(\tilde{n}, \tilde{x}) &\leq \alpha_1 - 2\sum_{s\in S}\sum_{z\in\mathcal{Y}:\phi_s(z)\in\mathcal{Y}} \tilde{v}^*_{s,z}(\tilde{n}_z - \psi_s(z)\tilde{n}_{\phi_s(z)}) - 2\sum_{s\in S}\sum_{z\in\mathcal{Y}:\phi_s(z)\in\mathcal{Z}\setminus\mathcal{Y}} \tilde{v}^*_{s,z}(\tilde{n}_z - \psi_s(z)X) - 2x\tilde{v}^* \\
&= \alpha_1 - 2\sum_{s\in S}\sum_{z\in\mathcal{Y}} \tilde{v}^*_{s,z} w_{s,z}(\tilde{n}, x) - 2\tilde{v}^* x, \\
&= \alpha_1 - \max\left(2\sum_{s\in S}\sum_{z\in\mathcal{Y}} \tilde{v}^*_{s,z} w_{s,z}(\tilde{n}, x), 2\tilde{v}^* x\right)
\end{aligned}
$$

Thus we get,

$$
D(\tilde{n}, \tilde{x}) \leq \alpha_1 - x \min_{c\in C} \sum_{s\in S} \frac{\delta_s}{4} p_{s,c}.
$$

Hence, for any $(\tilde{n}, x)$ such that $x > (\alpha_1 + \epsilon)\min_{c\in C}\sum_{s\in S}\frac{\delta_s}{4}p_{s,c}$, we have $D(\tilde{n}, \tilde{x}) \leq -\epsilon$.

We also have that

$$
D(\tilde{n}, \tilde{x}) \leq \alpha_1 - 2\sum_{s\in S}\frac{\delta_s}{4}\max_{z\in\mathcal{Y}} w_{s,z}(\tilde{n}, x).
$$

Thus,

$$
D(\tilde{n}, \tilde{x}) \leq \alpha_1 - 2\left(\min_{s\in S}\frac{\delta_s}{4}\right)\sum_{s\in S}\max_{z\in\mathcal{Y}} w_{s,z}(\tilde{n}, x) \leq \alpha_1 - 2\left(\min_{s\in S}\frac{\delta_s}{4}\right)\max_{z\in\mathcal{Y}}\sum_{s\in S} w_{s,z}(\tilde{n}, x).
$$

Now suppose that $x \leq \alpha_2 \triangleq (\alpha_1 + \epsilon)\min_{c\in C}\sum_{s\in S}\frac{\delta_s}{4}p_{s,c}$. Then, if we show that $\max_{z\in\mathcal{Y}}\sum_{s\in S} w_{s,z}(\tilde{n}, x) \to \infty$ as $|\tilde{n}|_\infty \to \infty$, then we have that $D(\tilde{n}, \tilde{x}) \leq -\epsilon$ a positive integer K' such that $|\tilde{n}|_\infty > K'$. We now show that $\sum_{s\in S}\max_{z\in\mathcal{Y}} w_{s,z}(\tilde{n}, x) \to \infty$ as $|\tilde{n}|_\infty \to \infty$.

Let $z^* = \arg\max_{z\in\mathcal{Y}} n_z$. Then we have

$$
\begin{aligned}
\sum_{s\in S} w_{s,z*}(\tilde{n}, x) &\leq \sum_{s}(n_{z^*} - \psi_s(z)\min(\alpha_2, n_{z^*})) \\
&= |S|n_{z^*} - \min(\alpha_2, n_{z^*})\sum_s \psi_s(z)
\end{aligned}
$$

which tends to infinity because

$$
\begin{aligned}
\sum_s \psi_s(z) &= \sum_s\sum_c z^*_c(1 - p_{s,c}) = |S| - \sum_s\sum_c z^*_c p_{s,c} \\
&\leq |S| - \max_c \sum_s z^*_c p_{s,c} \leq |S| - \max_c z^* c \min_c \sum_s p_{s,c} \\
&\leq |S| - \frac{1}{|C|}\min_c \sum_s p_{s,c} < |S|.
\end{aligned}
$$

Thus, there exist positive constants K and ϵ such that if $x > K$ or if $|\tilde{n}|_\infty > K$ then $D(\tilde{n}, \tilde{x}) \leq -\epsilon$.

Let $\mathcal{A} \triangleq \{(\tilde{n}, \tilde{x}) : \max(|\tilde{n}|_\infty, x) \leq K\}$. Then, using a version of Lyapunav-Foster Theorem from [34], we have that, from any state $(\tilde{n}, \tilde{x})$ such that $|\tilde{n}| + x < \infty$, the expected time to return to $\mathcal{A}$, i.e., $\tau_{\mathcal{A}}(\tilde{n}, \tilde{x})$ is finite. Further,

$$
T \triangleq \sup_{(\tilde{n},\tilde{x})\in\mathcal{A}} \tau_{\mathcal{A}}(\tilde{n}, \tilde{x}) < \infty.
$$

Thus, starting with any state in $\mathcal{A}$, we return to it in a finite expected time. We will be done if we show that expected time to return to state $(0, 0)$ is also finite. We do this as follows. Fix a constant $\beta > 0$. Since there exists

s such that $\min_c p_{s,c} > 0$, we have that for any interval of time of size β the probability that no arrival happens in the this interval and that a task leaves the system is finite.

Suppose that system is in a state $(\tilde{n}, \tilde{x}) \in \mathcal{A}$ at time $t = 0$. Now consider renewal times $T_i, i = 0, 1, 2, \ldots,$, where $T_0 = 0$ and for each $i > 0$, T_i is defined as follows: T_i is equal to $T_{i-1} + \beta$ if indeed no arrival happens and a task leaves the system in the interval $[T_{i-1}, T_i)$, else T_i is the first time of return to $\mathcal{A}$ after T_{i-1}. Clearly $E[T_i]$ since T as defined above is finite. Further probability that a task leaves system in time $T_i - T_{i-1}$ is fine, say α. Thus, time for system emptying after first reaching $\mathcal{A}$ can be upper-bounded by sum of K geometric random variables with rate α. Thus expected time to return to state $(0, 0)$ is fine. Hence, the system is stable.

Now suppose that the system is stable. Then, the necessary conditions can be shown to hold by the ergodicity of the system, and letting $v_{s,z}$ for each s, z to be the long-term fraction of times a server s attempts a task in N_z. □

8.4 Proof of Lemma 8.2

Upon rearrangement of terms in the expression of $f(\Theta)$ we obtain

$$f(\Theta)/2 = -\sum_{s} \sum_{z \in \mathcal{Y}: \phi_s(z) \in \mathcal{Y}} \theta_{s,z}(n_z - \psi_s(z')n_{\phi_s(z)}) - \sum_{s} \sum_{z \in \mathcal{Y}: \phi_s(z) \in \mathcal{Z} \setminus Y} \theta_{s,z}(n_z - \psi_s(z')x) - x\theta.$$

By using the definition of weights $w_{s,z}$, we obtain

$$f(\Theta)/2 = -\sum_{s} \sum_{z \in \mathcal{Y}} \theta_{s,z} w_{s,z}(\tilde{n}, x) - x\theta \geq -\sum_{s} \left(\max_{z \in \mathcal{Y}} w_{s,z}(\tilde{n}, x) \right) \sum_{z \in \mathcal{Y}} \theta_{s,z} - x\theta.$$

Thus,

$$\begin{aligned}
&f(\{v_{sz}\}_{s \in S, z \in \mathcal{Y}} \cup \{\min_{s} \delta_s/4\})/2 \\
&\geq -\sum_{s} \left(\max_{z \in \mathcal{Y}} w_{s,z}(\tilde{n}, x) \right) \sum_{z \in \mathcal{Y}} v_{s,z} - x \min_{c} \sum_{s \in S} (\delta_s/4) p_{sc} \\
&\geq -\sum_{s} (\mu_s - \delta_s/2) \max_{z \in \mathcal{Y}} w_{sz}(\tilde{n}, x) - x \min_{c} \sum_{s \in S} (\delta_s/4) p_{s,c} \\
&\geq -\mathbf{1}\left\{ \sum_{s} \max_{z \in \mathcal{Y}} w_{s,z}(\tilde{n}, x)(\mu_s - 3\delta_s/4) \geq x(\min_{c} \sum_{s} (\mu_s - 3\delta_s/4) p_{s,c}) \right\} \sum_{s} \max_{z \in \mathcal{Y}} w_{s,z}(\tilde{n}, x)(\mu_s - 3\delta_s/4) \\
&- \mathbf{1}\left\{ \sum_{s} \max_{z \in \mathcal{Y}} w_{s,z}(\tilde{n}, x)(\mu_s - 3\delta_s/4) < x(\min_{c} \sum_{s} (\mu_s - 3\delta_s/4) p_{s,c}) \right\} x \min_{c} \sum_{s} (\mu_s - 3\delta_s/4) p_{s,c} \\
&= f\left(\{\hat{v}^*_{sz}\}_{s \in S, z \in \mathcal{Y}} \cup \hat{v}^* \right)/2.
\end{aligned}$$

Hence, the lemma holds. □

8.5 Proof of Corollary 4.4

Proof. For this system we have $\mathcal{Z} = \{z', z''\}$ where $z'_c = \frac{1}{2}$ for each $c \in C$, and $z''_c = \mathbf{1}\{c = c_2\}$. The flow conservation constraints can be given as follows:

$$\lambda = \sum_{s} v_{s,z'}, \text{ and } \sum_{s} v_{s,z'} \psi_s(z') + \sum_{s} v_{s,z''} \psi_s(z'') = \sum_{s} v_{s,z''}$$

Suppose $a \geq \frac{1}{2}$. There exists an $\epsilon > 0$ such that $\lambda = \frac{3a(1-\epsilon)}{a+1}$. It can be checked that $\{\nu_{sz}\}_{s,z}$ where

$$\nu_{s_2,z'} = 1-\epsilon, \nu_{s_2,z''} = 0, \nu_{s_1,z'} = \frac{2a-1}{a+1}(1-\epsilon), \nu_{s_1,z''} = \frac{2-a}{a+1}(1-\epsilon)$$

and $\{\delta_s\}_{s\in S}$ where $\delta_s = \epsilon$ for each s satisfies sufficient conditions of Theorem 4.1.

Now suppose $a < \frac{1}{2}$. There exists an $\epsilon > 0$ such that $\lambda = 2a(1-\epsilon)$. It can be checked that $\{\nu_{s,z}\}_{s,z}$ where

$$\nu_{s_2,z'} = 2a(1-\epsilon), \nu_{s_2,z''} = 0, \nu_{s_1,z'} = 0, \nu_{s_1,z''} = (1-\epsilon)$$

and $\{\delta_s\}_{s\in S}$ where $\delta_s = \epsilon$ for each s satisfies sufficient conditions of Theorem 4.1.

The result then follows from the proof of Theorem 4.1 by taking $\mathcal{Y}$ as $\mathcal{Z}$. □

ACKNOWLEDGMENTS

The authors would like to thank Marc Lelarge of SAFRAN for valuable discussions.

REFERENCES

[1] Agrawal, Shipra and Goyal, Navin. Analysis of thompson sampling for the multi-armed bandit problem. In *Proceedings of the 25th Conference on Learning Theory*, 2012.

[2] Alresaini, M., Sathiamoorthy, M., Krishnamachari, B., and Neely, M.J. Backpressure with adaptive redundancy (bwar). In *Proc. IEEE INFOCOM*, pp. 2300–2308, March 2012.

[3] Asmussen, Søren. *Applied probability and queues.* Springer Science & Business Media, 2nd edition, 2003.

[4] Auer, Peter, Cesa-Bianchi, Nicolò, and Fischer, Paul. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2): 235–256, 2002.

[5] Bimpikis, Kostas and Markakis, Mihalis G. Learning and hierarchies in service systems. *Unpublished manuscript*, 2015.

[6] Brémaud, Pierre. *Markov chains: Gibbs fields, Monte Carlo simulation, and queues*, volume 31. Springer Science & Business Media, 2013.

[7] Bui, L.X., Srikant, R., and Stolyar, A. A novel architecture for reduction of delay and queueing structure complexity in the back-pressure algorithm. *IEEE/ACM Transactions on Networking*, 19(6):1597–1609, Dec 2011.

[8] Cui, Y., Yeh, E.M., and Liu, R. Enhancing the delay performance of dynamic backpressure algorithms. *IEEE/ACM Transactions on Networking*, 2015.

[9] Dai, J. G. On positive harris recurrence of multiclass queueing networks: A unified approach via fluid limit models. *The Annals of Applied Probability*, 5(1):49–77, 1995.

[10] Gao, Chao, Lu, Yu, and Zhou, Dengyong. Exact exponent in optimal rates for crowdsourcing. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pp. 603–611, 2016.

[11] Georgiadis, Leonidas, Neely, Michael J., and Tassiulas, Leandros. Resource allocation and cross-layer control in wireless networks. *Foundations and Trends in Networking*, 1(1):1–144, 2006.

[12] Gittins, John, Glazebrook, Kevin, and Weber, Richard. *Multi-armed bandit allocation indices.* John Wiley & Sons, 2011.

[13] Gurvich, Itay and Whitt, Ward. Scheduling flexible servers with convex delay costs in many-server service systems. *Manufacturing & Service Operations Management*, 11(2):237–253, 2009.

[14] Ho, Chien-Ju and Vaughan, Jennifer Wortman. Online task assignment in crowdsourcing markets. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI'12, pp. 45–51, 2012.

[15] Ji, Bo, Joo, Changhee, and Shroff, N.B. Delay-based back-pressure scheduling in multihop wireless networks. *IEEE/ACM Transactions on Networking*, 21(5):1539–1552, Oct 2013.

[16] Johari, Ramesh, Kamble, Vijay, and Kanoria, Yash. Know your customer: Multi-armed bandits with capacity constraints. abs/1603.04549, 2016. URL http://arxiv.org/abs/1603.04549.

[17] Karger, David R., Oh, Sewoong, and Shah, Devavrat. Budget-optimal task allocation for reliable crowdsourcing systems. *Operations Research*, 62(1):1–24, 2014.

[18] Kleinberg, Jon and Sandler, Mark. Convergent algorithms for collaborative filtering. In *Proceedings of the 4th ACM Conference on Electronic Commerce*, 2003.

[19] Kleinberg, Jon and Sandler, Mark. Using mixture models for collaborative filtering. In *Proceedings of the 36th annual ACM Symposium on Theory of Computing*, 2004.

[20] Lai, T.L and Robbins, Herbert. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4 – 22, 1985.

[21] Maguluri, Siva Theja, Burle, Sai Kiran, and Srikant, R. Optimal heavy-traffic queue length scaling in an incompletely saturated switch. *SIGMETRICS Perform. Eval. Rev.*, 44(1):13–24, June 2016.

[22] Mandelbaum, Avishai and Stolyar, Alexander L. Scheduling flexible servers with convex delay costs: Heavy-traffic optimality of the generalized cμ-rule. *Operations Research*, 52(6):836–855, 2004.

[23] Massoulié, Laurent. Structural properties of proportional fairness: Stability and insensitivity. *Annals of Applied Probability*, 17(3): 809–839, 2007.

[24] Massoulié, Laurent and Xu, Kuang. On the capacity of information processing systems. In *29th Annual Conference on Learning Theory*, New York, United States, June 2016.

[25] Mehta, A. and Panigrahi, D. Online matching with stochastic rewards. In *IEEE 53rd Annual Symposium on Foundations of Computer Science*, pp. 728–737, Oct 2012.

[26] Mehta, Aranyak, Waggoner, Bo, and Zadimoghaddam, Morteza. *Online Stochastic Matching with Unequal Probabilities*, pp. 1388–1404. 2015.

[27] Neely, M. J., Modiano, E., and Rohrs, C. E. Dynamic power allocation and routing for time varying wireless networks. In *Proc. IEEE INFOCOM*, 2003.

[28] Robert, Philippe. Stochastic networks and queues, stochastic modelling and applied probability series, vol. 52, 2003.

[29] Rybko, Aleksandr Nikolaevich and Stolyar, Alexander L. Ergodicity of stochastic processes describing the operation of open queueing networks. *Problemy Peredachi Informatsii*, 28(3):3–26, 1992.

[30] Shah, Nihar B., Zhou, Dengyong, and Peres, Yuval. Approval voting and incentives in crowdsourcing. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pp. 10–19, 2015.

[31] Shah, Virag, de Veciana, Gustavo, and Kesidis, George. A stable approach for routing queries in unstructured p2p networks. *IEEE/ACM Transactions on Networking*, 24(5):3136–3147, Oct 2016.

[32] Stern, David H., Herbrich, Ralf, and Graepel, Thore. Matchbox: Large scale online bayesian recommendations. In *Proceedings of the 18th International Conference on World Wide Web*, 2009.

[33] Tassiulas, L. and Ephremides, A. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transactions on Automatic Control*, 37:1936–1948, 1992.

[34] Tweedie, R. L. Criteria for classifying general markov chains. *Advances in Applied Probability*, 8(4):737–771, 1976.

[35] Ying, Lei, Shakkottai, Sanjay, Reddy, Aneesh, and Liu, Shihuan. On combining shortest-path and back-pressure routing over multihop wireless networks. *IEEE/ACM Transactions on Networking*, 19(3):841–854, June 2011. ISSN 1063-6692.

[36] Zhang, Yuchen, Chen, Xi, Zhou, Dengyong, and Jordan, Michael I. Spectral methods meet em: A provably optimal algorithm for crowdsourcing. *J. Mach. Learn. Res.*, 17(1):3537–3580, January 2016.