Elastic Job Scheduling with Unknown Utility Functions

Xinzhe Fu
LIDS, Massachusetts Institute of Technology
USA

Eytan Modiano
LIDS, Massachusetts Institute of Technology
USA

ABSTRACT

We consider a bipartite network consisting of job schedulers and parallel servers. Jobs arrive at the schedulers following stochastic processes with unknown arrival rates, and get routed to the servers, which execute the jobs with unknown service rates. The jobs are *elastic*, as their "size", i.e., the amount of service needed for their completion, is determined by the schedulers. After a job finishes execution, some utility is obtained where the utility value depends on the job's size through some underlying concave utility function. We consider the setting where the utility functions are unknown apriori, while a noisy observation of the utility value of each job is obtained upon its completion. Our goal is to design a policy that makes job-size and routing decisions to maximize the total utility obtained by the end of the time horizon T. We measure the performance of a policy by regret, i.e., the gap between the expected utility obtained under the policy and that under the optimal policy. We first establish an upper bound on the regret of a generic policy, that consists of the cumulative difference in utility between the job-size decisions of the policy and the solution to a static optimization problem, and the total backlog of unfinished jobs at the end of the time horizon. We then propose a policy that simultaneously controls the cumulative utility difference and backlog of unfinished jobs, and achieves an order optimal regret of $\tilde{O}(\sqrt{T})$. Our policy solves the elastic job scheduling problem by extending the Stochastic Convex Bandit Algorithm to handle unknown and stochastic constraints, and making routing decisions based on the Join-the-Shortest-Queue rule. It also presents a principled approach to extending algorithms for zeroth-order convex optimization to the settings with unknown and stochastic constraints.

1 INTRODUCTION

Job scheduling is a class of problems that study schedule construction and resource allocation to jobs over a set of machines to optimize for various performance objectives. In many job scheduling applications, the jobs to be scheduled are *elastic*, that is, the arriving jobs do not have a pre-determined size or duration but instead their "sizes" are determined by the system scheduler [1, 2], and the utility gained from job completion depends on the "allocated" job size [3, 4]. A typical example is training tasks for machine learning models. The training process of many machine learning models (e.g. deep neural network) involves iterative procedures such as gradient descent [5, 6]. The model's performance resulting from the training (i.e., utility of the job) depends on the number of iterations completed (i.e., size of the job) [7]. Thus, it is possible to take advantage of such elasticity to dynamically determine the sizes of incoming jobs to achieve considerable gain in terms of the overall performance, as described in [8, 9].

An important element in the scheduling of elastic jobs is the jobs' utility functions, i.e., the underlying relationship between the job size and the corresponding utility. Such utility functions are usually non-decreasing with respect to the job size, and are (approximately) concave, which reflects, for example, the observation that model performance increases with more training time while the marginal gain in performance diminishes with training time [10]. Moreover, the utility functions are often unknown apriori, but function values corresponding to job-size decisions can be observed. Again, using machine learning training as an example, the training curve is typically unknown in advance, but the model performance of a certain training time can be observed after a corresponding training task is completed [11]. While monotonicity and concavity have often been utilized to design scheduling algorithms with provable guarantees [3, 4], the unknown nature of the utility function has been overlooked by most works in the literature, which assume the utility functions to be known beforehand.

In this paper, we study the problem of elastic job scheduling with unknown utility functions. We consider a discrete-time system of a bipartite network with K job schedulers and a set S of parallel servers. There are K classes of jobs, with jobs of each class arriving at their corresponding job scheduler according to a discrete-time stochastic process with mean rate λ_k. Each class is associated with some unknown underlying utility function f_k. At every time t, each job scheduler k decides for each incoming job j, the job size x_j and its designated server, and then routes the job to the queue of its designated server. After a job j of class k finishes its service at its designated server, we obtain a utility of $f_k(x_j)$ and receive a noisy observation of the function value $f_k(x_j) + \epsilon_j$, where ϵ_j is a zero-mean noise and assumed to be independent for different jobs. The goal is to design a policy that makes job-size and routing (choice of designated server) decisions based on observed information, in order to maximize the total utility obtained from jobs completed by the end of the time horizon T. We adopt regret, which is equal to the difference between the utility obtained by the optimal policy and that of our policy, as the performance metric and propose a policy with order-optimal regret.

2 MODEL AND PROBLEM FORMULATION

Consider a discrete-time system with a set of job schedulers and a set of parallel servers that form a bipartite network. We use $U = \{u_1, \ldots, u_K\}$ to denote the set of schedulers and $S = \{s_1, \ldots, s_M\}$ to denote the set of servers. Each scheduler u_k is connected to a subset $S_{u_k} \subseteq S$ of servers. Each server has a buffer that stores the jobs to be processed. There are K classes of elastic jobs in the system, where jobs of class k arrive at scheduler u_k and are sent to a server in S_{u_k} for execution. At each time slot t, a set $A_k(t)$ of class k jobs with $|A_k(t)| = a_k(t)$ arrive at scheduler u_k. For each job j, its corresponding scheduler determines its size $x_j \in [0, B]$, which is the workload it will add to the server and can be interpreted as its resource requirement. The scheduler then sends job j to the buffer of a server $s_j \in S_{u_k}$ for execution, which we will refer to as j's designated server. Server s_m's service rate

at time t is denoted by $c_m(t)$. Each server executes the jobs in a non-preemptive fashion. We assume that for each k, $a_k(t)'s$ form a sequence of i.i.d. bounded positive integer random variables, and for each m, $c_m(t)$'s is a sequence of i.i.d. bounded non-negative random variables. We assume, $\mathbb{E}[a_k(t)] = \lambda_k$, $\mathbb{E}[c_m(t)] = \mu_m$ and $1 \leq a_k(t), \lambda_k, c_m(t), \mu_m \leq C$. We will refer to the jobs' arrival rates λ_k's and the servers' service rates c_m's, as *network statistics*. In this work, we consider the setting where the network statistics are unknown, but the realizations of arrivals and service are observable.

Each class k is associated with some underlying utility function f_k that characterizes the relationship between the size and the utility value obtained from jobs of class k. The underlying utility functions are unknown, but we can receive noisy zeroth-order feedback on the utility functions. Specifically, after the server finishes executing a job of size x of class k, we observe $f_k(x) + \epsilon$ and obtain a utility of $f_k(x)$, where ϵ is a zero-mean bounded random noise. The noise values of different jobs are independent. We assume that for each job class k, its underlying utility function f_k is monotonically non-decreasing, concave, and Lipschitz-continuous.

We study a finite-horizon elastic job scheduling problem. Given a time horizon T, we seek a scheduling policy that determines the size of arriving jobs and their designated servers such that the total utility obtained from the jobs that are completed in T time slots is maximized. Let Π^* be the set of all policies, including the ones that know the underlying utility functions and network statistics. For a policy π, let $U(\pi, T)$ be the total utility obtained under policy π, which is defined as the sum of utility obtained from jobs that have been completed by the end of the time horizon T. Note that $U(\pi, T)$ is a random variable, the randomness of which comes from job arrivals, service rates, noisy utility observations and the (possible) inherent randomness in the scheduling policy π. Instead of directly using $U(\pi, T)$, we adopt the notion of regret as the measure of the quality of scheduling policies, which is defined as

$$R(\pi, T) = \sup_{\pi^* \in \Pi^*} \mathbb{E}[U(\pi^*, T)] - \mathbb{E}[U(\pi, T)].$$

3 MAIN RESULTS

Consider the following optimization problem $\mathcal{P}$ where Λ is the stability region of the network:

$$\mathcal{P}: \max_{\{x_k\}} \sum_{k=1}^{K} \lambda_k f_k(x_k) \tag{1}$$

$$\textbf{s.t.} \quad (x_1, \ldots, x_K) \in \Lambda, \tag{2}$$

$$x_k \in [0, B], \quad \forall k. \tag{3}$$

Intuitively, the optimization problem characterizes the job scheduling problem with full information in steady state. The decision variables $\{x_k\}$ can be interpreted as the steady-state size of jobs of class k. As the objective function of $\mathcal{P}$ is concave while the feasibility region is a convex set, it follows that $\mathcal{P}$ is a convex optimization problem.

Our first theorem establishes that the expected utility of any policy in Π^* is upper-bounded by the optimal value of the $\mathcal{P}$ times the time horizon T.

Theorem 1. $\sup_{\pi^* \in \Pi^*} \mathbb{E}[U(\pi^*, T)] \leq T \cdot OPT(\mathcal{P})$.

Based on theorem 1, we can essentially transform the elastic job scheduling problem to a convex optimization problem with (zeroth-order) bandit feedback. The state-of-art algorithm for such problems is the Stochastic Convex Bandit Algorithm (SCBA) proposed in [12]. However, since our problem has stochastic constraints, i.e. the set Λ involves unknown parameters associated with the statistics of stochastic processes, SCBA cannot be directly applied. To address this challenge, we first construct an equivalent formulation of $\mathcal{P}$,

$$\tilde{\mathcal{P}}: \max_{\boldsymbol{x}} F(\boldsymbol{x}) := \sum_{k=1}^{K} \lambda_k f_k(x_k) - C(L+1)\Delta(\boldsymbol{x}, \Lambda)$$

$$\textbf{s.t.} \quad x_k \in [0, B], \quad \forall k,$$

where $\Delta(\boldsymbol{x}, \Lambda)$ denotes the l_1 distance of $\boldsymbol{x}$ to the set Λ. $\tilde{\mathcal{P}}$ does not involve stochastic constraints and can be shown to be equivalent to $\mathcal{P}$. However, we do not have unbiased observations of the objective function of $\tilde{\mathcal{P}}$, which was essential to SCBA. We thus further propose a procedure that utilizes observations available in the setting of the elastic job scheduling problem to construct confidence intervals around the true utility values, which can play essentially the same role in SCBA. Combined with the confidence interval construction procedure, we can extend SCBA to solve the elastic job scheduling problem and achieves the order-optimal $\tilde{O}(\sqrt{T})$-regret.

Theorem 2. *There exists a policy that achieves $\tilde{O}(\sqrt{T})$-regret for the elastic job scheduling problem.*

ACKNOWLEDGMENTS

This work was funded by NSF grants CNS-1524317, NSF CNS-1907905 and by Office of Naval Research (ONR) grant award N00014-20-1-2119.

REFERENCES

[1] A. Wierman and M. Nuyens. "Scheduling despite inexact job-size information." in *Proceedings of the ACM SIGMETRICS*, 2008.
[2] S. T. Maguluri and R. Srikant. "Scheduling jobs with unknown duration in clouds."in *IEEE/ACM Transactions On Networking*, Vol. 22, No. 6, pp: 1938-1951, 2013.
[3] Y. Zheng, B. Ji, N. Shroff, and P. Sinha. "Forget the deadline: Scheduling interactive applications in data centers." in *IEEE International Conference on Cloud Computing*, pp: 293-300, 2015.
[4] Z. Zheng, N. Shroff. "Online multi-resource allocation for deadline sensitive jobs with partial values in the cloud." in *IEEE INFOCOM*, pp: 1-9, 2016.
[5] Y. Bao, Y. Peng, C. Wu, and Z. Li. "Online job scheduling in distributed machine learning clusters." in *IEEE INFOCOM*, pp: 495-503, 2018.
[6] Y. Peng, Y. Bao, Y. Chen, C. Wu, and C. Guo. "Optimus: an efficient dynamic resource scheduler for deep learning clusters." in *Proceedings of the Thirteenth EuroSys Conference*, pp: 1-14, 2018.
[7] A. Harlap, A. Tumanov, A. Chung, G. R. Ganger, and P. B. Gibbons. "Proteus: agile ml elasticity through tiered reliability in dynamic resource markets." in *Proceedings of the Twelfth European Conference on Computer Systems*, pp: 589-604, 2017.
[8] A. Or, H. Zhang, and M. Freedman. "Resource elasticity in distributed deep learning." in *Proceedings of Machine Learning and Systems*, Vol. 2, pp: 400-411, 2020.
[9] H. Zhang, L. Stafman, A. Or, and M. Freedman. "Slaq: quality-driven scheduling for distributed machine learning." in *Proceedings of the Symposium on Cloud Computing*, pp: 390-404. 2017.
[10] M. Jeon, S. Venkataraman, A. Phanishayee, J. Qian, W. Xiao, and F. Yang. "Analysis of large-scale multi-tenant GPU clusters for DNN training workloads." in *USENIX Annual Technical Conference*, pp: 947-960, 2019.
[11] A. Klein, S. Falkner, J. Springenberg, and F. Hutter. "Learning curve prediction with Bayesian neural networks." 2016.
[12] A. Agarwal, D. P. Foster, D. Hsu, S. M. Kakade, and A. Rakhlin. "Stochastic convex optimization with bandit feedback." in *SIAM Journal on Optimization*, Vol. 23, No. 1, pp: 213-240, 2013.