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 $O(\sqrt{T})$. Our policy solves the elastic job scheduling problem by extending the Stochastic Convex Bandit Algorithm to handle unknown and stochastic constraints, and makes routing 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.

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 $\lambda_k$. 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 $\epsilon_j$ 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
which is defined as the sum of utility obtained from jobs that have variables whose values of different jobs are independent. We assume that for each value obtained from jobs of class $k$, $\lambda_k$ is the arrival rate, $\mu_k$ is the service rate, $\gamma_k$ is the utility obtained from the jobs that are completed in $T$ time slots is $\Delta(x, \Lambda)$, $\tilde{P}$ is a policy 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 achieve the order-optimal $O(\sqrt{T})$-regret.

**Theorem 2.** There exists a policy that achieves $O(\sqrt{T})$-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**


