Contract Scheduling with Distributional and Multiple Advice

SPYROS ANGELOPOULOS, CNRS and LIP6-Sorbonne University, France
MARCIN BIENKOWSKI, University of Wroclaw, Poland
CHRISTOPH DÜRR, CNRS and LIP6-Sorbonne University, France
BERTRAND SIMON, CNRS / IN2P3 Computing Center, France

Additional Key Words and Phrases: Contract scheduling, Distributional advice, Multiple advice, Learning-augmented algorithms

ACM Reference Format:
Spyros Angelopoulos, Marcin Bienkowski, Christoph Dürr, and Bertrand Simon. 2018. Contract Scheduling with Distributional and Multiple Advice. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)*. ACM, New York, NY, USA, 3 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

This paper focuses on scheduling *contract algorithms*. These algorithms require an allowed computation time as part of their input, and, after that time, return a solution. The more time is allotted, the better the solution quality. By consecutively running such an algorithm with increasing computation times, we obtain an interruptible *anytime* algorithm: after any duration (larger than some threshold), the execution can be interrupted without warning and a result must be output. The objective is of course to get a solution of better quality when the interruption happens late, in order to take advantage of the available computation time. For instance, consider the schedule where the ith execution of the contract is allotted a time 2^i, for $i \in \mathbb{Z}$. The schedule starts by infinitesimally small executions to ensure that one contract is executed before the interruption. For any interruption time $t > 0$, the longest execution by the contract algorithm has lasted a duration at least $t/4$. This factor 4 between the performance achieved and the best performance in hindsight (execute a single contract for a time t) is called the *acceleration ratio*. More formally, define a schedule as a sequence $X = (x_i)_{i\in\mathbb{Z}}$ where x_i represents the length of the ith execution. Let $\ell(X, T)$ represent the length of the last contract terminated by X at time T. The acceleration ratio of X is defined as

$$(X) = \sup_T \frac{T}{\ell(X,T)}.$$

It is known that the best acceleration possible equals 4, achieved by the doubling schedule described above [1]. Contract scheduling has then been studied in more complex scenarios including the resort to multiple processors, the need to complete multiple instances, or the presence of soft deadlines.

Authors' Contact Information: Spyros Angelopoulos, spyros.angelopoulos@lip6.fr, CNRS and LIP6-Sorbonne University, Paris, France; Marcin Bienkowski, University of Wroclaw, Wroclaw, Poland; Christoph Dürr, christoph.durr.fr, CNRS and LIP6-Sorbonne University, Paris, France; Bertrand Simon, bertrand.simon@cnrs.fr, CNRS / IN2P3 Computing Center, Villeurbanne, France.

In this paper, we study this problem under the lens of learning-augmented algorithms [2]. In this recent field, we assume that the scheduler has access to a *prediction* about the interruption time. This prediction typically comes from patterns learned on previous instances, and has an inherent quality, unknown to the scheduler. We focus on the objective to obtain the best acceleration ratio assuming the prediction is perfect (i.e., the *consistency*), while maintaining an acceleration of 4 even if the prediction is totally inaccurate (i.e., the *robustness*). Previous works [3] have shown that if the prediction received is a single value representing the interruption time, the best possible consistency equals 2. It is obtained by shifting the doubling algorithm so that the prediction meets the end of an execution.

A predictor may however be unable to provide such an information accurately, based on past instances for example. We consider here two models which allow some flexibility on the information given. In the first model, the prediction comes in the form of a probability distribution of the interruption time. In the second one, the prediction provides multiple possible interruption times. In each setting, we establish the best possible consistency achievable while retaining a robustness factor equal to 4. We then analyze numerically the performance of our solutions in various scenarios.

2 DETAILED TECHNICAL RESULTS

For a given $\lambda \in [0,1)$, define the schedule $X(\lambda) = (2^{i-\lambda})_{i\in\mathbb{Z}}$. The following proposition shows that, as we focus on 4-robust schedules, it suffices to consider the set of schedules $\cup_{\lambda\in[0,1)}\{X(\lambda))\}$.

Proposition 2.1. *For any $\lambda \in [0,1)$, $X(\lambda)$ is 4-robust. Conversely, every 4-robust schedule must belong in the class $\cup_{\lambda\in[0,1)}\{X(\lambda)\}$.*

In the *distributional* model, we assume that the predictor provides a probability distribution μ describing the estimated interruption time. We define the consistency of a schedule X given prediction μ as the expected maximum length of a contract in hindsight divided by the expected length of the last contract executed by X:

$$c(X,\mu) = \frac{_{T\sim\mu}[T]}{_{T\sim\mu}[\ell(X,T)]}.$$

We first show that, for any distribution μ, it is possible to obtain a consistency better than 4 simply by choosing the best schedule among a set of evenly shifted schedules. For any n, let S_n denote the following collection of n schedules $X_0 \ldots, X_{n-1}$, defined as $X_j = (2^{i-j/n})_{i\in\mathbb{Z}}$.

Theorem 2.2. *For any $n \in \mathbb{N}^*$, there exists a 4-robust schedule in S_n that has consistency at most $4n \cdot (2^{1/n} - 1)$.*

We further show that this bound is the best possible.

Theorem 2.3. *For any $n \in \mathbb{N}^*$, there exists a distributional prediction for which every collection C_n that consists of n 4-robust schedules cannot contain a schedule of consistency smaller than $4n \cdot (2^{1/n} - 1)$.*

Noting that the limit of $4n \cdot (2^{1/n} - 1)$ equals $4\ln 2$, we can also achieve this limit with a continuous distribution. For any $D > 0$, define the probability distribution μ_D over $[D; 2D]$ by the density function $f_D(x) = 2D/x^2$.

Theorem 2.4. *For any $D > 0$ and every 4-robust schedule X, we have $c(X,\mu_D) \geq 4\ln 2$.*

Therefore, for any D, μ_D is a prediction for which the best 4-robust schedule achieves the worst consistency. We show that such result still exhibits some smoothness: if the actual interruption time distribution is close to μ_D (taking the Earth Mover Distance as a metric), then the acceleration ratio is close to $4 \ln 2$. Such a feature would not be present if all the probability mass of the prediction is located at a single point, as a slight decrease of the interruption time leads to an execution which is not completed.

In the *multiple advice* model, the predictor provides a set of potential interruption times P. We define the consistency of a schedule X assuming an adversary selects the interruption time among P which leads to the highest acceleration ratio:

$$c(X, P) = \sup_{\tau \in P} \frac{\tau}{\ell(X, \tau)}.$$

As we focus on schedules of the form $X(\lambda)$, we show that we can reduce the analysis by basically assuming all points of P belong to some interval $[D; 2D]$. Then, we construct a method determining which point of P should coincide with the end of a contract. This yields the following tight bound.

Theorem 2.5. *For any P, there exists a schedule computable in time $O(k^2)$ that has consistency at most $2^{2-\frac{1}{k}}$, where k is the size of P. Furthermore, this bound is tight, in that there exists a prediction P such that every 4-robust schedule has consistency at least $2^{2-\frac{1}{k}}$.*

REFERENCES

[1] Stuart J. Russell and Shlomo Zilberstein. *Composing real-time Systems.* Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI), 1991.

[2] Thodoris Lykouris and Sergei Vassilvitskii. *Competitive Caching with Machine Learned Advice.* Journal of the ACM, 2021.

[3] Spyros Angelopoulos and Shahin Kamali. *Contract Scheduling with Predictions*, J. Artif. Intell. Res., 2023.