Contract Scheduling with Predictions

Spyros Angelopoulos, Shahin Kamali

Contract Scheduling with Predictions*

Spyros Angelopoulos spyros.angelopoulos@lip6.fr
CNRS and LIP6–Sorbonne University
4 place Jussieu, Paris, France 75252

Shahin Kamali kamalis@yorku.ca
Department of Electrical Engineering and Computer Science
York University, Toronto, Canada

Abstract

Contract scheduling is a general technique that allows the design of systems with interruptible capabilities, given an algorithm that is not necessarily interruptible. Previous work on this topic has assumed that the interruption is a worst-case deadline that is unknown to the scheduler. In this work, we study new settings in which the scheduler has access to some imperfect *prediction* in regards to the interruption. In the first setting, which is inspired by recent advances in learning-enhanced algorithms, the prediction describes the time that the interruption occurs. The second setting introduces a new model in which predictions are elicited as responses to a number of binary queries. For both settings, we investigate tradeoffs between the robustness (i.e., the worst-case performance of the schedule if the prediction is generated adversarially) and the consistency (i.e., the performance assuming that the prediction is error-free). We also establish results on the performance of the schedules as function of the prediction error.

1. Introduction

One of the central objectives in the design of intelligent systems is the provision of *anytime* capabilities. In particular, several applications such as medical diagnostic systems and motion planning algorithms require that the system outputs a reasonably efficient solution given any unavoidable constraints on the computation time. *Anytime algorithms* offer such a tradeoff between computation time and quality of the output (Zilberstein, 1996). Namely, in an anytime algorithm, the quality of output improves gradually as the computation time increases. This class of algorithms was introduced first in (Boddy & Dean, 1994) in the context of time-depending planning, as well as in (Horvitz, 1988) in the context of flexible computation, and has since found several applications in the development of real-time and computationally-intensive systems (Zilberstein, 1996; Zilberstein & Russell, 1993).

(Russell & Zilberstein, 1991; Zilberstein & Russell, 1996) introduced a useful distinction between two different types of anytime algorithms. On the one hand, there is the class of *contract* algorithms, which describes algorithms that are given the amount of allowable computation time (i.e, the intended query time) as part of the input. However, if the algorithm is interrupted at any point before this "contract time" expires, the algorithm may output a result that is meaningless. A typical example is algorithms based on dynamic programming

*. Preprint accepted to *Journal of Artificial Intelligence Research*. A preliminary version of this work appeared in the Proceedings of the 35th AAAI Conference on Artificial Intelligence (Angelopoulos & Kamali, 2021).

(DP); if the algorithm fails to fill the entire DP table, the output may be entirely useless. On the other hand, the class of *interruptible* algorithms consists of algorithms whose allowable running time is not known in advance, and thus can be interrupted (queried) at any given point throughout their execution. Such algorithms include versions of local search, e.g., simulated annealing and hill climbing.

Although less flexible than interruptible algorithms, contract algorithms typically use simpler data structures, and are thus often easier to implement and maintain (Bernstein, Finkelstein, & Zilberstein, 2003). Hence a natural question arises: how can one convert a contract algorithm to an interruptible equivalent, and at which cost? This question can be addressed in a problem-specific manner, depending on the algorithm at hand; however, there is a simple, iterative-deepening technique that applies to any possible contract algorithm, and consists of repeated executions of the contract algorithm with increasing runtimes (also called *lengths*). For example, consider a *sequencing*, or *schedule* (as most commonly called in the literature) of executions of the contract algorithm in which the i-th execution has length 2^i. Assuming that an interruption occurs at time t, then the above schedule guarantees the completion of a contract algorithm of length at least $t/4$, for any t. The factor 4 measures the performance of the schedule and quantifies the penalty due to the repeated executions.

More formally, given a contract algorithm A, a *schedule* X is defined by an increasing sequence $(x_i)_{i\geq 1}$ in which x_i is the length of the i-th execution of A. For simplicity, we call the i-th execution of A in X the i-th *contract*, and we call x_i its *length*. The *acceleration ratio* of X, denoted by $\texttt{acc}(X)$, relates an interruption T to the length of the largest contract that has completed by time T in X, which we denote by $\ell(X, T)$, and is defined as

$$\texttt{acc}(X) = \sup_T \frac{T}{\ell(X, T)} \tag{1}$$

At an intuitive level, the acceleration ratio describes a trade-off between the speed of the processor in which the contracts are scheduled, and the resilience of the system to interruptions. Namely, by executing the schedule X to a processor of speed equal to $\texttt{acc}(X)$, one obtains a system that is as efficient as a single execution of a contract algorithm that knows when the interruption will occur, but runs in a unit-speed processor. Hence, the objective is to obtain a schedule that minimizes this measure. Figure 1 illustrates the definition.

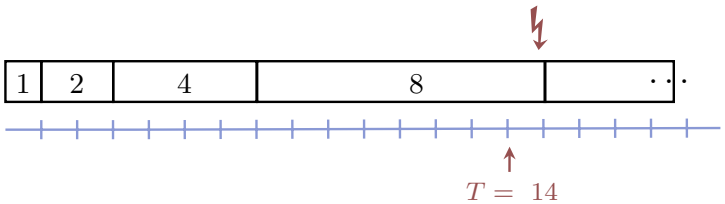


Figure 1: An illustration of the schedule $X = (2^i)$. Suppose that an interruption occurs at time $T = 14$, i.e., before the completion of the contract of length 8. The longest completed contract has length 4, namely $\ell(X, 14) = 4$, and the performance of the schedule at the time of this interruption is measured by the ratio $14/4$.

Contract scheduling has been studied in a variety of settings related to AI. It has long been known that the schedule $X = (2^i)$ attains the optimal acceleration ratio which is equal to 4 (Russell & Zilberstein, 1991). Optimal schedules in multi-processor systems were obtained in (Bernstein, Perkins, Zilberstein, & Finkelstein, 2002). The generalization in which there are more than one problem instances associated with the contract algorithm was first studied in (Zilberstein, Charpillet, & Chassaing, 2003), in which optimal schedules were obtained for a single processor. The more general setting of multiple instances and multiple processors was first studied in (Bernstein et al., 2003) and later in (López-Ortiz, Angelopoulos, & Hamel, 2014). (Angelopoulos, López-Ortiz, & Hamel, 2008) considered the problem in which the interruption is not a fixed deadline, but there is a "grace period" within which the system is allowed to complete the execution of the contract. Measures alternative to the acceleration ratio were proposed and studied in (Angelopoulos & López-Ortiz, 2009). (Angelopoulos & Jin, 2019) studied contract scheduling in the setting in which the schedule is deemed "complete" once a contract reaches some prespecified end guarantees.

Contract scheduling is an abstraction of resource allocation under uncertainty, in a worst-case setting. As such, it has connections to other problems of a similar nature, such as the problem of *online searching* under the *competitive ratio* (Bernstein et al., 2003; Angelopoulos, 2015; Kupavskii & Welzl, 2019; Angelopoulos, 2021).

1.1 Contract scheduling with predictions

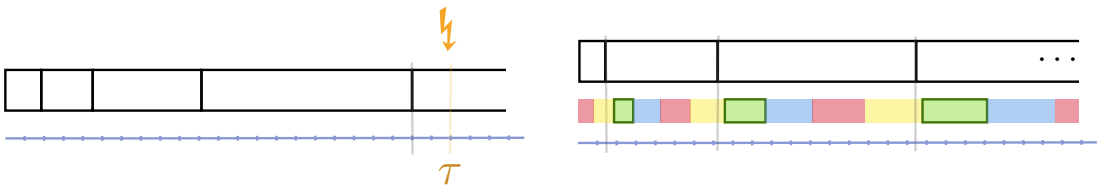
Previous work on contract scheduling has largely assumed that the interruption is unknown to the scheduler, and thus can be chosen adversarially, in particular right before a contract terminates. In practice, however, the scheduler could benefit from a certain *prediction* concerning the interruption. Consider the example of a medical diagnostic system. Here, the expert may know that the system will be likely queried around a *specific time*, (i.e., prior to a scheduled surgery). Another possible prediction may describe a partition of time in *intervals*, in which the system will likely be queried. Using again the example of the medical diagnostic system, it may be more likely that the consultation will be required over a weekday, than over a weekend.

We study two settings that capture the above scenarios. In the first setting, there is a prediction τ concerning explicitly the interruption T. This model is motivated by recent advances in *learning-enhanced online computation*, in which the online algorithm leverages a prediction concerning its input, namely the sequence of requests; see e.g. (Lykouris & Vassilvitskii, 2018) and (Purohit, Svitkina, & Kumar, 2018). In this setting, the prediction error η is naturally defined as the distance between the prediction and the actual interruption.

Our second setting describes a novel, *query-based* model for eliciting predictions: namely, the prediction is expressed in the form of responses to n *binary queries*, where n is a specified parameter, and the prediction error $\eta \leq 1$ is the fraction of erroneous responses. For example, a single binary query could be of the form "Will the interruption occur within a certain subset of the timeline?". This model combines aspects of query-based optimization, in which the algorithm recovers the solution to a problem by asking queries (as in clustering with noisy queries (Mazumdar & Saha, 2017), where a query asks whether two elements belong in the same cluster) and learning-enhanced optimization (as discussed above). Note,

however, that in our setting we do not make any probabilistic assumptions in regards to the query responses. In contrast, (Mazumdar & Saha, 2017) assumes that each query is erroneous with probability p that is known to the clustering algorithm. Queries can be either *static*, in that the order in which they are asked is decided ahead of time, or *adaptive* in that the next query to be asked can be a function of the responses to previous queries. For our negative results (lower bounds on the acceleration ratio) we will assume the full power of adaptive queries; this only strengthens the results (see e.g., Theorem 8). For our positive results (upper bounds on the acceleration ratio), we will consider either adaptive or static queries (see Theorems 10 and 11, respectively).

Figure 2 illustrates the ways in which the scheduler may leverage predictions towards improving the acceleration ratio. For the setting in which the prediction is the interruption time, the scheduler may choose to complete a large contract close, but prior to the predicted interruption time. For the query-based setting, we observe that n queries can induce a partition of the time in up to 2^n disjoint sets. Once again, the scheduler may choose to complete large contracts prior to the predicted set in which the interruption is likely to occur. We emphasize that this is only a high-level illustration of the concepts, since it does not take into account the significant complications due to the prediction error, which will be the central issue in the analysis presented in this work.



(a) interruption time as prediction

(b) query-based prediction

Figure 2: An illustration of the prediction models. In (a), the prediction is an estimate τ of the interruption time. Here, an efficient schedule would aim to complete a large contract somewhat earlier than τ. In (b), there is a query-based prediction using $n = 2$ binary queries; thus, the time is partitioned into $2^n = 4$ sets of intervals, with each color describing such a set. Suppose that the query responses indicate that the interruption takes place in the highlighted (green) set. An efficient schedule would aim to complete large contracts prior to the start time of each one of the green intervals.

To evaluate the performance of the schedule, and in line with the evaluation of learning-enhanced online algorithms, we are first interested in the two extreme situations, in regards to the prediction error. In the first extreme, we assume *adversarial* error, i.e., an adversary that knows the schedule, and can manipulate the prediction accordingly; in this case, the worst-case performance of the schedule is called *robustness*. In the second extreme, the prediction is perfect, i.e., *error-free*; in this case, the worst-case performance of the schedule is called *consistency*. In between these extremes, the acceleration ratio is, in general, a function of the prediction error, and one naturally aims to design schedules whose acceleration ratio degrades gracefully as function of the error.

While the algorithm (i.e., the scheduler) is always oblivious to the prediction error, there may be situations in which it may have access to some *upper bound* on the error, which can also be interpreted as the algorithm's desired, worst-case *tolerance* to errors. For example, in a medical diagnostic system, the expert may wish to wait only up to a certain amount of time beyond the predicted time at which the system will be queried. We will denote this upper bound by H, and we will refer to schedules with access to H as *H-aware*, otherwise we call the schedule *H-oblivious*. This parameter is also related to the concept of *weak predictions*, used recently in the analysis of learning-augmented online algorithms (see, e.g., online knapsack with frequency predictions (Im, Kumar, Qaem, & Purohit, 2021), in which the prediction is in fact an upper bound on the size of items that appear online). Such an assumption is also commonly made in the analysis of games with a lying responder, namely in *Reńyi-Ulam games*, see e.g., (Rivest, Meyer, Kleitman, Winklmann, & Spencer, 1980), in which an upper bound on the erroneous responses is assumed to be known.

1.2 Contribution

We first consider the setting in which the prediction τ is the interruption time T. This prediction comes with an *error* $\eta \in [0, 1]$ such that $T \in [\tau(1-\eta), \tau(1+\eta)]$. We first obtain a schedule that is *Pareto-optimal*, namely attains the best trade-off between consistency and robustness, by establishing a reduction from an online problem known as *online bidding* (Chrobak & Kenyon-Mathieu, 2006). This allows us to use a Pareto-optimal algorithm of (Angelopoulos, Dürr, Jin, Kamali, & Renault, 2020), and obtain a schedule with the same ideal performance. There are two complications: this schedule is fairly complex, and cannot tolerate *any* errors. To overcome these issues, we give a simpler schedule (based on geometrically increasing contract lengths) with the same robustness and consistency, and which is thus also Pareto-optimal. We then show how to extend this schedule to the realistic setting in which $\eta \neq 0$, and we complement the positive results with several lower bounds on the performance of any schedule, both in regards to H-aware and H-oblivious schedules.

In the second part, we study the query-based setting, in which the prediction is in the form of responses to n binary queries, for some given parameter n, i.e., we would like to combine the advice of n binary experts. We first address the issue of consistency/robustness tradeoffs. We show an information-theoretic lower-bound equal to $2^{1+\frac{1}{2^n}}$ on the best consistency one can hope to achieve, assuming a desired robustness equal to 4 (i.e., the best-possible robustness). We also present and analyze, for any given robustness $r \geq 4$ a schedule that makes efficient use of the query responses, and matches the lower bound for $r = 4$. We then define and analyze a family of schedules, parameterized by the extreme value of the error H that each schedule can tolerate. There are several challenges here: the analysis must incorporate parameters such as the error η, the upper bound H, the number of queries n and the desired robustness r. Moreover, we need to define queries that are realistic and have a practical implementation. To this end, we give an explicit implementation of each query as a *partition* query of the form "Does interruption T belong to $\mathcal{T}$?", where $\mathcal{T}$ is a suitably defined subset of the timeline. Our approach, in both settings, is to consider a parallel version of the problem, in which the response to queries can help us choose the best among a collection of candidate schedules.

For both prediction models, we complement the theoretical analysis with an experimental evaluation of our schedules. The results demonstrate that the empirical improvements are in line with the theoretical analysis. They also demonstrate that predictions can lead to substantial improvements, namely to acceleration ratios that are well below the best-possible in the prediction-free setting.

It is worth underlining that unlike "natural" online optimization problems studied in works such as (Lykouris & Vassilvitskii, 2018) and (Purohit et al., 2018), contract scheduling under the acceleration ratio poses certain novel challenges. Most notably, it is not the case that the performance improves monotonically as the error decreases. To see this, consider an interruption T, a prediction τ for T, and a schedule X for prediction τ. Suppose that a contract finishes right before τ in X: this is intuitively bad, because even with a very small error, it is possible that X barely misses to complete its largest contract by time T. But it is also possible that if the error is very large, T happens to occur right after another contract terminates in the schedule. This is a "best-case" scenario for the schedule: it completes a contract right on time. This observation exemplifies the type of difficulties we face. Another difficulty is that there may exist schedules that are Pareto optimal for the pair of consistency and robustness, but whose performance falls back to the worst-case acceleration ratio for *any* non-zero error. Such schedules are clearly undesirable, which is another challenge we must overcome.

The query-based prediction model we introduced in this work can be applicable to other optimization problems under uncertainty. See, e.g. (Angelopoulos, Kamali, & Zhang, 2022), for a recent application in the context of online financial optimization, following the conference version of this work.

1.3 Other related work

There are several recent works that study algorithms with machine-learned predictions in a status of uncertainty. Examples include online rent-or-buy problems with multiple expert predictions (Gollapudi & Panigrahi, 2019), queuing systems with job service times predicted by an oracle (Mitzenmacher, 2020), online algorithms for metrical task systems (Antoniadis, Coester, Elias, Polak, & Simon, 2020), online makespan scheduling (Lattanzi, Lavastida, Moseley, & Vassilvitskii, 2020), graph exploration (Eberle, Lindermayr, Megow, Nölke, & Schlöter, 2022) and the Steiner tree problem (Xu & Moseley, 2022), to mention only some representative results. See also the survey (Mitzenmacher & Vassilvitskii, 2020). In particular, studies of Pareto-efficient algorithms with respect to consistency-robustness tradeoffs have become prominent recently in the context of online optimization problems with untrusted predictions, see e.g., (Angelopoulos et al., 2020; Sun, Lee, Hajiesmaili, Wierman, & Tsang, 2021; Wei & Zhang, 2020; Li, Yang, Qu, Shi, Yu, Wierman, & Low, 2022; Lee, Maghakian, Hajiesmaili, Li, Sitaraman, & Liu, 2021).

Concerning contract scheduling, the work that is closest to ours is (Zilberstein et al., 2003), in which there is stochastic information about the interruption, and the objective is to optimize the expected quality of the output upon interruption. The optimal scheduling policy in (Zilberstein et al., 2003) is based on a Markov decision process, hence no closed-form solution is obtained. More importantly, their schedule does not provide worst-case

guarantees (i.e., a bound on the robustness), but only average-case guarantees for the given distribution, which is also assumed to be known.

2. Preliminaries

A contract schedule is defined by a sequence $X = (x_i)_{i\geq 1}$ of contract lengths, or *contracts*, where x_i is the *i-th contract* in X. We will always denote by T the time at which an interruption occurs. We will make the standing assumption that an interruption can occur only after a unit time has elapsed, otherwise no schedule has a finite acceleration ratio. With no prediction on T, the worst-case acceleration ratio of X is given by (1) as explained in Section 1; this is the *robustness* of X, which we denote by r_X, or simply r, if the schedule is implied. With a prediction, the acceleration ratio of X is simply defined as $T/\ell(X,T)$. Given a prediction, the *consistency* of X is its acceleration ratio assuming $\eta = 0$. We will say that a schedule has *performance* (r, s) if it has robustness r and consistency s, and we will call such as schedule r-robust and s-consistent. A *Pareto-optimal* schedule attains the best-possible tradeoff between the consistency and the robustness.

Given a schedule $X = (x_i)_{i\geq 1}$ (where recall that $(x_i)_{i\geq 1}$ is an increasing sequence), it is easy to see that the worst-case interruptions occur infinitesimally prior to the completion of a contract (Russell & Zilberstein, 1991). Hence the following useful expression:

$$r_X = \sup_{i\geq 1} \frac{\sum_{j=1}^{i} x_j}{x_{i-1}}, \tag{2}$$

where x_0 is defined to be equal to -1.

The class of *exponential* schedules describes schedules in which the i-th contract has length a^i, for some fixed a, which we call the *base* of the schedule. For several variants of contract scheduling, there are efficient schedules in this class. The robustness of an exponential schedule with base a is equal to $a^2/(a-1)$ (Zilberstein et al., 2003), and for $a = 2$ the corresponding schedule has optimal robustness 4 (Russell & Zilberstein, 1991). Let

$$c_r = \frac{r - \sqrt{r^2 - 4r}}{2} \text{ and } b_r = \frac{r + \sqrt{r^2 - 4r}}{2}, \tag{3}$$

be the two roots of the function $x^2/(x-1) - r$, then from the discussion above, it follows that for any given $r \geq 4$, an exponential schedule with base $a \in [c_r, b_r]$ has robustness at most r. This fact will be useful in our analyses.

In the *online bidding* problem (Chrobak & Kenyon-Mathieu, 2006), we seek an increasing sequence $X = (x_i)_{i\geq 1}$ of positive numbers (called *bids*) of minimum *competitive ratio*, defined formally as

$$\sup_{u\geq 1} \frac{\sum_{j=1}^{i} x_j}{u} \ : x_{i-1} < u \leq x_i, \tag{4}$$

where u is the target value. In words, we seek a strategy for submitting bids, given some unknown target (or threshold) u, and we pay a cost equal to the sum of all bids up the first bid that is at least as large as u. The competitive ratio of the strategy is the maximum ratio of this cost divided by the target u.

Without predictions, online bidding is equivalent to contract scheduling: given an increasing sequence $X = (x_i)$, both its acceleration ratio and its competitive ratio can be described by (2). Consider now a variant of online bidding, in which some prediction is given concerning the target u. We say that a bidding sequence has performance (r, s) with a given prediction if it has robustness r and consistency s with respect to its competitive ratio.

3. Interruption time as prediction

We first study the setting in which the prediction τ describes the interruption time T. The prediction comes with an *error* $\eta \in [0, 1]$, defined as follows. If $T \geq \tau$, then we define η to be such that $T/\tau = 1 + \eta$, and if $T \leq \tau$, then we define η to be such that $T/\tau = 1 - \eta$. In the former case, we will say that the error is *positive*, otherwise we will say that the error is *negative*. Regardless of the sign of error, we have that $T \in [\tau(1-\eta), \tau(1+\eta)]$.

The parity of the error will help us establish more precise performance guarantees in terms of the theoretical analysis of the acceleration ratio. Note that we assume, implicitly, that $\tau \leq 2T$. There are two ways to justify this assumption. First, if τ is too large, in comparison to T, then the prediction is not helpful, especially in the context of real-time applications. Second, as explained in Section 1.2, one cannot hope for schedules that are universally efficient for all possible values of prediction error.

We will also study settings in which the error η is bounded by a quantity $H \leq 1$ which may or may not be known to the schedule; we thus distinguish between H-oblivious and H-aware schedules. Note that if η is bounded by H then

$$\tau(1-H) \leq \tau(1-\eta) \leq T \leq \tau(1+\eta) \leq \tau(1+H).$$

We will first consider the ideal case in which the prediction is either error-free (hence the consistency is evaluated for $\eta = 0$), or it is adversarially generated (hence the robustness is the worst-case acceleration ratio). We establish a simple, yet useful reduction between our problem and the online bidding problem with predictions.

Theorem 1. *Suppose that for every $r \geq 4$, there is a sequence for the online bidding problem that has performance (r, s) for prediction equal to the target u. Then there is a contract schedule with the same guarantees for the setting in which the prediction is the interruption, and vice versa.*

Proof. We first give the reduction from contract scheduling to bidding. Namely, we have a prediction τ for the interruption, and let $X = (x_i)_{i \geq 1}$ an (r, s)-competitive bidding sequence for prediction of a target $u = \tau$. Let m be the smallest index such that $x_m \geq \tau$ in X. We can assume, without loss of generality, that $x_m = \tau$, otherwise, by scaling down the bids by a factor x_m/τ we obtain a new sequence that is no worse than X, in both its robustness and its consistency. From the definition of consistency for X, we have that

$$\sum_{i=1}^{m} x_i \leq s \cdot x_m = s \cdot \tau. \tag{5}$$

Moreover, from the definition of r-robustness we have that

$$\sum_{j=1}^{i} x_j \leq r \cdot x_{i-1}, \text{ for all } i \geq 1. \tag{6}$$

Consider now the schedule X_τ in which the lengths of the contracts are defined by the sequence $X_\tau = (x'_i)$, in which $x'_i = x_i/s$. Then in X_τ, we have that the contract x'_m is completed by time $(\sum_{i=1}^{m} x_i)/s \leq \tau$ (from (5)), and the consistency of the schedule is at most $\frac{\sum_{i=1}^{m}(x'_i)}{x'_m} \leq s$, again from (5). Last, note that since X satisfies (6), so does X_τ, and thus the schedule must be r-robust.

The reduction in the opposite direction follows along the same lines, by scaling up the contract lengths so as to obtain the bids. □

From (Angelopoulos et al., 2020), there is a Pareto-optimal bidding sequence, which satisfies the conditions of Theorem 1 with $s = c_r$. This implies the existence of a Pareto-optimal schedule for contract scheduling. In what follows, we denote this schedule by X^*_τ.

Corollary 2. *For every $r \geq 4$, there is a contract schedule X^*_τ that has performance (r, c_r), and this is Pareto-optimal.*

We also obtain the following corollary, which will be useful in establishing the theoretical guarantees in the more general setting of non-zero prediction error.

Corollary 3. *For any r-robust schedule X, and any $\epsilon > 0$, there exists t_0 such that for all $t \geq t_0$, we have that $\ell(X, t) \leq t(1+\epsilon)/c_r$. Moreover, for any $\epsilon > 0$, there exists i_0 such that if $x_i = \ell$, with $i \geq i_0$, then the completion time of x_i is at least $c_r\ell(1-\epsilon)$.*

Proof. The corollary follows from previous studies of the linear recurrence relation $\sum_{j=1}^{i} x_j \leq rx_{i-1}$, and in particular, Corollary 3 in (Angelopoulos, 2021). See Appendix for details. □

There are two issues that one needs to address. The first issue is that the schedule obtained using the reduction to online bidding has a fairly complex statement, because the bidding algorithm in (Angelopoulos et al., 2020) is quite complex, namely described in terms of a recurrence relation. We can give instead an explicit, and more intuitive exponential schedule that has the same performance, hence is also Pareto-optimal.

In particular, consider the geometric schedule $G = (b_r^i)_{i=1}^{\infty}$, where b_r is defined in (3). Then there exists a scaling factor $\gamma < 1$ such that in the schedule $(\gamma b_r^i)_{i=1}^{\infty}$, there is a contract that completes at time equal to τ. For example, suppose that $\tau = 100$ and $r = 4.5$, which gives $b_r = 3$. Then we have $b_r^5 = 243$, and hence $\gamma = 100/243$.

We will show that this simple schedule, which we will denote by X^*_τ, also has performance (r, c_r), and thus is also Pareto-optimal, from Corollary 2. First, from the discussion in Section 2, we know that the schedule $(b_r^i)_{i\geq 1}$ is r-robust, and so is then X^*_τ due to the scaling of all contracts by γ. Moreover, from the definition of γ, at time τ, X^*_τ completes a contract of length γb_r^m, for some m, and the consistency of the schedule is at most

$$\sum_{i=1}^{m} \frac{\gamma b_r^i}{\gamma b_r^m} \leq \frac{b_r}{b_r - 1} = c_r,$$

where the last equality follows directly from the definitions of b_r and c_r.

The second, and more significant issue, is that as in the case of the online bidding algorithm of (Angelopoulos et al., 2020), in the presence of *any* error $\eta \neq 0$, the acceleration ratio of X^*_τ becomes as bad as its robustness r. This is because if $T = \tau - \epsilon$, i.e., if the error is negative, and infinitesimally small, then the longest contract in X^*_τ barely misses its completion. That is, the schedule makes a very inefficient use of its resources, even when the error is negligible.

We will thus show how to adapt X^*_τ in order to obtain a more realistic schedule that is robust to prediction errors. The idea is to allow some "buffer" so that the schedule can tolerate mispredictions as a function of the buffer size. More precisely, for any $p \in (0,1)$, consider the schedule $X^*_{\tau(1-p)}$. The following lemma gives an upper bound on the performance of this parameterized, and H-oblivious schedule.

Lemma 4. *For any $p \in (0,1)$, and $r \geq 4$, $X^*_{\tau(1-p)}$ is r-robust and has consistency $\min\{\frac{c_r}{1-p}, r\}$. It also has acceleration ratio at most $\min\{\frac{c_r(1+\eta)}{1-p}, r\}$ for positive error, at most $\min\{\frac{c_r(1-\eta)}{1-p}, r\}$ if η is negative error with $\eta \leq p$, and at most r, in every other case.*

Proof. First, note that by construction the schedule is guaranteed to be r-robust, so neither its acceleration ratio, nor its robustness can exceed r. With prediction τ, $X^*_{\tau(1-p)}$ completes a contract of length $l = \frac{\tau(1-p)}{c_r}$ by time T (this follows from the statement of the schedule, and Corollary 2). By definition, the acceleration ratio of the schedule is at most $\frac{T}{l}$; moreover, for positive error η we have that $T = \tau(1+\eta)$, whereas for negative error $\eta \leq p$ we have that $T = \tau(1-\eta)$. For no error, we have that $T = \tau$ (for which the consistency is evaluated). Combining the above observations yields the lemma. See Figure 3 for an illustration. □

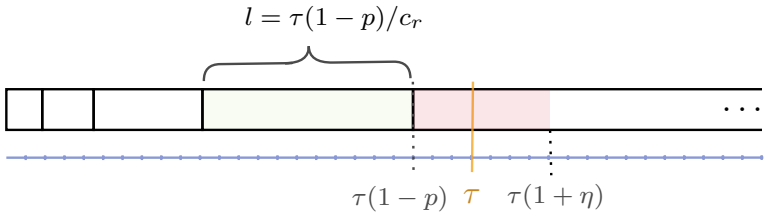


Figure 3: An illustration of Lemma 4. Assuming negative error at most p, the interruption occurs no earlier than $\tau(1-p)$, and no later than $\tau(1+\eta)$ for positive error η. In the former case, in particular, a contract of length $l = \frac{\tau(1-p)}{c_r}$ is guaranteed to be completed by the interruption time.

The above result provides a tradeoff between the acceleration ratio of $X^*_{\tau(1-p)}$, and the range in which it is sufficiently good, as a function of the error. To illustrate this, consider the case of negative error: If p is relatively small, then the schedule has good acceleration ratio for relatively small η ($\eta < p$), which however can become as large as r, for a relatively big range of error, i.e, for $\eta > p$.

We now argue that these tradeoffs are unavoidable, in any r-robust and H-oblivious schedule X with prediction τ. Recall that $\ell(X, \tau)$ denotes the largest contract completed

in the schedule by time τ in X, and let $p \in [0,1]$ be such that $\tau(1-p)$ is the completion time of this contract. From Corollary 3 we know that for arbitrarily small $\epsilon > 0$, and any sufficiently large T (and hence τ, as well) we have that $\ell(X,\tau) \leq \tau(1-p)(1+\epsilon)/c_r$, hence for negative error $\eta \leq p$, the acceleration ratio is at least $\frac{c_r(1-\eta)}{(1-p)(1+\epsilon)}$, therefore the consistency of the schedule is at least $\frac{c_r}{(1-p)(1+\epsilon)}$. Moreover, there exists $x > 0$ such that at time $\tau(1+x)$, the largest completed contract does not exceed $\ell(X,\tau)$. Hence for positive error $\eta < x$, the acceleration ratio is at least $\frac{c_r(1+\eta)}{(1-p)(1+\epsilon)}$. Last, since the schedule is H-oblivious, T can occur at points right before a contract terminates, for all contracts that completed before τ. In this latter case, the acceleration ratio will inevitably be as large as r, as T becomes large.

For these reasons, we will next consider *H-aware* schedules in which $\eta \leq H$ and H is known. A natural schedule then is $X^*_{\tau(1-H)}$, in which the buffer p is determined by H. Its performance is described in the following lemma, whose proof follows similarly to Lemma 4, by setting $p = H$.

Lemma 5. *$X^*_{\tau(1-H)}$ is r-robust, and has acceleration ratio at most $\min\{\frac{c_r(1+\eta)}{(1-H)}, r\}$ for positive error, and at most $\min\{\frac{c_r(1-\eta)}{(1-H)}, r\}$ for negative error.*

Since $\eta \leq H$, we have $\mathtt{acc}(X^*_{\tau(1-H)}) \leq \min\{\frac{c_r(1+H)}{1-H}, r\}$. The next lemma shows that H can take values in a certain range, as function of c_r, for which no other r-robust schedule can be better.

Lemma 6. *For any H that satisfies the condition $\frac{1+H}{1-H} < \sqrt{\frac{c_r+1}{c_r}}$, the acceleration ratio of any H-aware r-robust schedule is at least $\min\{\frac{c_r(1+H)}{1-H}, r\}$.*

Proof. By way of contradiction, let X denote an H-aware schedule that has acceleration ratio at most $\min\{\frac{c_r(1+H)}{1-H}, r\}$. Then given prediction τ, X must complete by time $\tau(1-H)$ a contract, say x, of length at least

$$\frac{\tau(1-H)^2}{c_r(1+H)}.$$

From Corollary 3, the completion time of x must be at least

$$c_r \cdot \frac{\tau(1-H)^2}{c_r(1+H)}(1-\epsilon) = \frac{\tau(1-H)^2}{1+H}(1-\epsilon),$$

for arbitrarily small $\epsilon > 0$. We now claim that x is also the largest contract completed by time $\tau(1+H)$ in X. By way of contradiction, suppose that there is a contract y that follows x, and which completes by time $\tau(1+H)$. Note that y must be at least as big as x. Then it must be that

$$\frac{\tau(1-H)^2}{1+H}(1-\epsilon) + \frac{\tau(1-H)^2}{c_r(1+H)} \leq \tau(1+H),$$

and since ϵ can be arbitrarily small, we arrive at a contradiction on the assumption on H. Thus, if $T = \tau(1+H)$ (i.e., for positive $\eta = H$, the largest contract completed is x, and thus the acceleration ratio is at least $c_r(\frac{1+H}{1-H})^2 \geq c_r\frac{1+H}{1-H}$. □

We can also show that there is an even larger range for H than that of Lemma 6, for which no other schedule can *dominate* $X^*_{\tau(1-H)}$, in the sense that no schedule can have as good an acceleration ratio as $X^*_{\tau(1-H)}$ on all possible values of $\eta \leq H$, and strictly better for at least one such value.

Lemma 7. *For any H such that $\frac{1+H}{1-H} < \frac{c_r+1}{c_r}$, no r-robust H-aware schedule dominates $X^*_{\tau(1-H)}$.*

Proof. By way of contradiction, suppose that there exists an r-robust schedule X that dominates $X^*_{\tau(1-H)}$. Then X must complete a contract, say x, of length at least $\tau(1-H)/c_r$, by time $\tau(1-H)$. Using the same arguments as in the proof of Lemma 6, it follows that X does not complete any contract bigger than x by time $\tau(1+H)$. This implies that for all possible values of error, X has the same acceleration ratio as $X^*_{\tau(1-H)}$, which contradicts the dominance assumption. □

Example. To put the above results into perspective, let us consider the case $r = 4$ (best-possible robustness). Then $c_r = 2$, and X^*_τ is 2-consistent, but can have acceleration ratio 4 for any $\eta \neq 0$. For given bound H, $X^*_{\tau(1-H)}$ has acceleration ratio at most $\min\{\frac{2(1+\eta)}{1-H}, 4\}$ for positive error, and at most $\min\{\frac{2(1-\eta)}{1-H}, 4\}$ for negative error. Thus, an absolute upper bound on its acceleration ratio is $\min\{\frac{2(1+H)}{1-H}, 4\}$, whereas its consistency is $\min\{\frac{2}{1-H}, 4\}$. For any $H < 0.101$, no 4-robust H-aware schedule has better acceleration ratio. Last, for $H < 0.2$, there is no 4-robust H-aware schedule that dominates $X^*_{\tau(1-H)}$.

4. Query-based predictions

In this section, we study the setting in which the prediction is in the form of responses to n binary queries $Q_1, \ldots, Q_n$, for some given n. For example, a query can be of the form "will the interruption occur after time $t = 100$?", or may be even more complex, e.g., "will the interruption occur in the set $\cup_{i=\text{odd}}[2^i, 2^{i+1}]$?". Hence, the prediction P can be viewed as an n-bit string, where the i-th bit is the response to Q_i. Recall also that the prediction error $\eta \in [0, 1]$ is defined as the fraction of erroneous bits in P. We will assume, for simplicity, that the total number of erroneous bits, namely ηn, is an integer.

Our approach in this setting is as follows. Let $\mathcal{X}$ be a set of r-robust schedules. The prediction P will help choose a good schedule from this set. For positive results, we need to define $\mathcal{X}$, and show how the prediction can help us choose an efficient schedule from it; moreover the prediction must have a practical interpretation (i.e., cannot come from an overly powerful oracle), and must tolerate errors. For negative (i.e., impossibility) results, we need to show that for *any* choice of 2^n r-robust schedules in $\mathcal{X}$, one cannot guarantee consistency below a certain bound. Note that in this scheme, all schedules in $\mathcal{X}$ must be r-robust, because any schedule in $\mathcal{X}$ can be chosen, if the prediction is adversarially generated.

To offer some intuition behind our overall approach, let us refer back to Figure 2(b). The main idea is to think of each partition of the timeline induced by the query responses as giving rise to a separate contract scheduling problem, which implies that with n query responses, there will be up to 2^n potential schedules to choose from, each performing well

for one of the 2^n partitions (observe that in Figure 2(b), we have $n = 2$, thus 4 partitions). The challenge here is to use queries that are robust to errors, in that even if some of the responses are erroneous (and thus even if we chose a schedule that is efficient for some partition other than the one in which the interruption belongs) we will not be too far from the best-possible schedule.

We will address several issues related to this setting. We begin, in Section 4.1 with the simpler, but still challenging setting in which there is no prediction error, and we show positive and negative results on the robustness/consistency tradeoff. In Section 4.2 we focus on the more realistic case in which the predictions are erroneous, and we give a schedule whose performance degrades gently as function of its tolerance to errors. In Section 4.3 we discuss issues related to the precise statements of the queries, from an implementation standpoint.

4.1 Consistency/robustness tradeoffs

We begin with a negative result, for the simple, but important case $r = 4$, i.e., for best-possible robustness. The following theorem establishes an information-theoretic lower bound on the consistency.

Theorem 8. *For any query-based prediction P of size n, any schedule with performance $(4, s)$ is such that $s \geq 2^{1+\frac{1}{2^n}}$.*

Proof. We first give an outline of the proof. With n binary queries, the prediction P can only help us choose a schedule from a class $\mathcal{X}$ of at most 2^n 4-robust schedules. Let $X_1, X_2, \ldots, X_{2^n}$ describe these schedules. By way of contradiction, suppose we could guarantee consistency $2^{1+\frac{1}{2^n}} - \delta$, with $\delta > 0$. We will show that there exists an ordering of these schedules with the following property, which we prove by induction: there is a set of $2^n - 1$ interruptions, $T_2, \ldots, T_{2^n}$ such that, for interruption T_i, with $i \in [2, 2^n]$, no schedule of rank at most $i+1$ in the ordering can guarantee consistency $2^{1+\frac{1}{2^n}}$. This means that for interruption T_{2^n}, no schedule in $\mathcal{X}$ can guarantee consistency $2^{1+\frac{1}{2^n}}$, a contradiction.

We now proceed with the technical details. Let $S = 2^{1+\frac{1}{2^n}}$. By way of contradiction, suppose there is a schedule Z, which is 4-robust, and which has consistency $S - \delta$, for some $\delta > 0$. We will rely on the following information-theoretic argument: with n-bit prediction, Z can only differentiate between a set $\mathcal{X}$ of 2^n schedules. Each schedule in $\mathcal{X}$ must be 4-robust, otherwise Z cannot be 4-robust either. Let $X_1, \ldots X_{2^n}$ denote the 2^n schedules in $\mathcal{X}$, and we denote by $x_{i,l}$ the length of the i-th contract in X_l. We also define $T_{i,l} = \sum_{j=1}^{i} x_{i,l}$ as the completion time of the i-th contract in X_l. We will say that for a given interruption T, Z *chooses* schedule X_l in $\mathcal{X}$ if the prediction P points to this schedule. We can assume that Z will always choose a schedule that has completed the *largest* contract completed by time T, among all schedules in $\mathcal{X}$; this only strengthens the claims.

Let us fix some index $i \in \mathbb{N}^+$. There exists a schedule in $\{X_2, \ldots, X_{2^n}\}$ that has completed the *largest* contract by time $T_{i,1}$ among all these schedules. Without loss of generality, we can assume that this schedule is X_2 (by re-indexing the schedules), and we denote by $\bar{i}_2$ the index of this largest contract in X_2. Inductively, for all $l \in [2, 2^n - 1]$, there has to be a schedule in $\{X_{l+1}, \ldots X_{2^n}\}$ which has completed the largest contract by time $T_{\bar{i}_l,l}$ among all these schedules. Again, without loss of generality, we can assume that

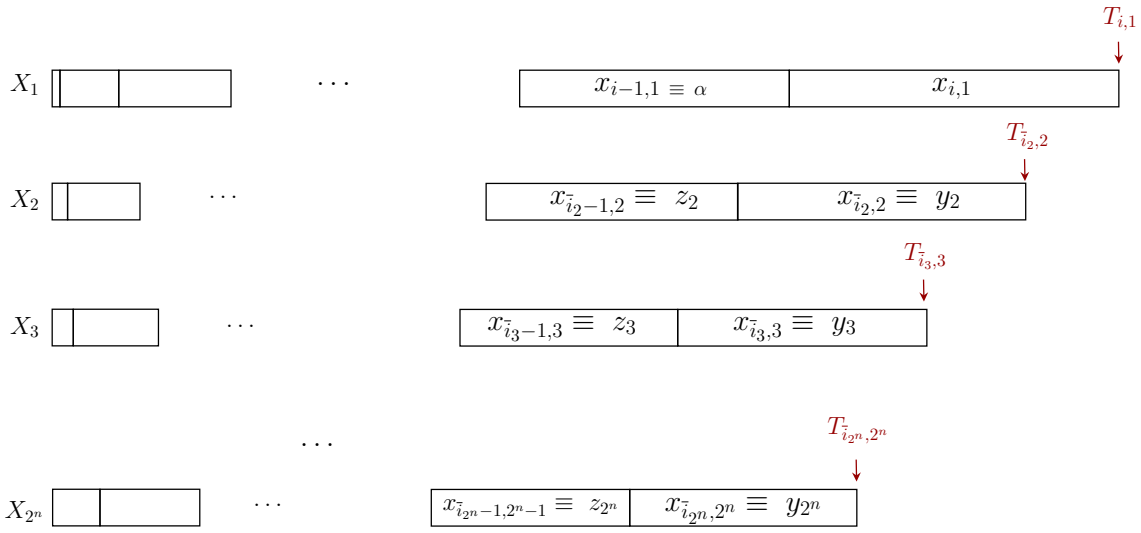


Figure 4: An illustration of some key concepts in the proof of Theorem 8 and Lemma 9.

this schedule is X_{l+1}, and we denote by $\bar{i}_{l+1}$ the index of this largest contract in X_{l+1}. We will say that $\mathcal{X}$ is *ordered for index* i if its schedules obey the above properties. Figure 4 illustrates the setting and the notation used in the proof.

We will use Corollary 3 with $r = 4$; in this case, the corollary states that for every 4-robust schedule $X = (x_i)_{i\geq 1}$, and every $\epsilon > 0$ there exists i_0 such that $\sum_{j=1}^{i} x_j \geq (2-\epsilon)x_i$, for all $i \geq i_0$. Hence, for every $\epsilon > 0$, we can then choose i_0 sufficiently large such that the conditions of Corollary 3 apply to X_1, but also to all schedules in $\mathcal{X}$. Consider then any fixed $i \geq i_0$, and let $\mathcal{X}$ be ordered for i. To simplify a bit the notation, let y_l be equal to $x_{\bar{i}_l,l}$. Then from Corollary 3 we have that

$$T_{\bar{i}_l,l} \geq (2-\epsilon)y_l, \text{ for all } l \in [1, 2^n], \text{ and } i \geq i_0. \tag{7}$$

Note that ϵ can be chosen to be arbitrarily small, for sufficiently large i_0. To simplify the proofs, in what follows we will assume that (7) holds with $\epsilon = 0$. We can make this assumption without affecting correctness, because $\delta > 0$ is fixed, and ϵ can be chosen arbitrarily smaller than δ.

The proof of the theorem will follow directly from Lemma 9, which we prove separately. This is because for $l = 2^n$, the lemma shows that Z cannot choose any schedule in $\mathcal{X}$ so as to guarantee consistency strictly less than S, a contradiction. □

Lemma 9. *Consider the setting described in the proof of Theorem 8, and let α be equal to $x_{i-1,1}$. Suppose that Z has consistency at most $S - \delta$, and that it is ordered for i. Then there exists $i \geq i_0$ such that for every $l \in [2, 2^n]$, it must be that $y_l \geq 2^{1-\frac{l-1}{2^n}}\alpha$. Moreover, for every $l \in [2, 2^n]$, if an interruption occurs at a time infinitesimally earlier than $T_{\bar{i}_l,l}$, then Z cannot choose any schedule in $\{X_1, \dots X_l\}$.*

Proof. We first simplify the notation a little. We will define z_l to be equal to $x_{\bar{i}_l-1,l}$, i.e., the contract immediately preceding y_l in X_l. Recall that α is defined to be equal to $x_{i-1,1}$.

This contract plays a special role in the proof, as we will see. Let $s = S - \delta$ denote the consistency of Z.

The proof of the lemma is by induction on l. We show the base case, namely $l = 2$, which gives some intuition for the proof of the inductive step, and we refer the reader to the appendix for the full proof of the inductive step.

First, we will prove the lower bound on y_2. Recall that X_1 is 4-robust; furthermore, we know that no schedule can be better than 4-robust. This implies that

$$\sup_{i \geq i_0} \frac{T_{i,1}}{x_{i-1,1}} = \sup_{i \geq i_0} \frac{T_{i,1}}{\alpha} \geq 4.$$

By way of contradiction, suppose that $y_2 < 2^{1-\frac{1}{2^n}}\alpha$. Consider an interruption infinitesimally earlier than $T_{i,1}$. There are two possibilities: either Z chooses X_1 or it chooses X_2 (since $\mathcal{X}$ is ordered for i, this is the best schedule among $X_2, \ldots, X_{2^n}$). In the former case, we have that

$$s \geq \frac{T_{i,1}}{\alpha},$$

whereas in the latter case we have that

$$s \geq \frac{T_{i,1}}{y_2} \geq \frac{T_{i,1}}{2^{1-\frac{1}{2^n}}\alpha}.$$

These two possible cases apply for all $i \geq i_0$. Hence we obtain that

$$s \geq \sup_{i \geq i_0} \frac{T_{i,1}}{2^{1-\frac{1}{2^n}}\alpha} \geq \frac{4}{2^{1-\frac{1}{2^n}}} = 2^{1+\frac{1}{2^n}} = S,$$

a contradiction.

Next, we will show that there exists an interruption such that Z cannot choose either X_1 or X_2. Recall that by definition, $T_{\bar{i}_2,2}$ is the completion time of y_2, hence from (7) we have that

$$T_{\bar{i}_2,2} \geq 2y_2 \geq 2 \cdot 2^{1-\frac{1}{2^n}}\alpha = 2^{2-\frac{1}{2^n}}\alpha.$$

Consider an interruption infinitesimally earlier than $T_{\bar{i}_2,2}$. Suppose, by way of contradiction, that Z chooses X_1. Then it must be that

$$\frac{T_{\bar{i}_2,2}}{x_{i-1,1}} \leq s < S \Rightarrow S > 2^{2-\frac{1}{2^n}},$$

a contradiction.

Next, suppose, again by way of contradiction, that Z chooses X_2. Note that by definition, the largest contract completed by the above interruption in X_2 is z_2, and since z_2 cannot exceed y_2, from (7) we obtain that

$$z_2 \leq \frac{T_{\bar{i}_2,2} - y_2}{2}.$$

Therefore, using again (7) we infer that

$$\begin{aligned} \frac{T_{\bar{i}_2,2}}{z_2} < S \Rightarrow S &> 2\frac{T_{\bar{i}_2,2}}{T_{\bar{i}_2,2} - y_2} \\ &\geq 4\frac{y_2}{T_{\bar{i}_2,2} - y_2} \geq 8\frac{y_2}{T_{\bar{i}_2,2}} \end{aligned} \tag{8}$$

However, we know that since X_1 is 4-robust, then $T_{i,1} \leq 4\alpha$. Since Z is ordered for i, we know that $T_{\bar{i}_2,2} \leq T_{i,1}$, thus $T_{\bar{i}_2,2} \leq 4\alpha$. Combining with the above inequality we obtain that

$$S > 2 \cdot y_2 \geq 2 \cdot 2^{1-\frac{1}{2^n}} = 2^{2-\frac{1}{2^n}}, \tag{9}$$

a contradiction. This concludes the base case. Please refer to the appendix for the full proof. □

We complement Theorem 8 with a positive result, which we establish in Theorem 10. Consider the set $\mathcal{X} = \{X_i, i \in [0, 2^n - 1]\}$ of schedules, in which $X_i = (x_{j,i})_{j \geq 1}$ is defined by $x_{j,i} = d^{j+\frac{i}{2^n}}$, for $d > 1$ that we will choose later[1]. In words, X_i is a near-exponential schedule with base d, and a scaling factor equal to $d^{\frac{i}{2^n}}$. The prediction P then chooses an index, in $[0, 2^n - 1]$, that determines a schedule in $\mathcal{X}$. We call Ideal the schedule obtained from $\mathcal{X}$ with prediction P.

We first illustrate the structure of $\mathcal{X}$, and explain the salient concepts behind the proof of the performance guarantees. One key fact we use in the proof, has to do with the fact that $\mathcal{X}$ has a "nice" structure, namely it is comprised by near-exponential schedules. Consider a time T right before the j-th contract of schedule i is about to complete, where $i \in [1, n-1]$. Then the largest contract among all schedules in $\mathcal{X}$ that has completed by time T is $x_{j,i-1}$ (i.e., a schedule of index one less in the cyclic order). If $i = 0$, then the largest contract that has completed by time T in $\mathcal{X}$ is $x_{j-1,n-1}$ (again, note that the schedule indexed $n-1$ has index one less than schedule 0 in the cyclic order). See Figure 5 for an illustration.

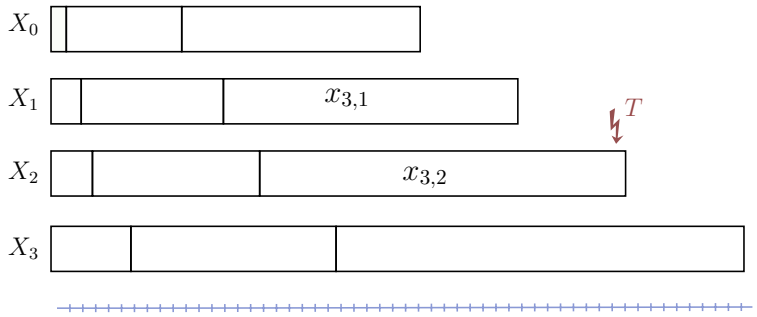


Figure 5: An illustration of some key concepts in the proof of Theorem 10. Here, $n = 2$, and $\mathcal{X}$ consists of 4 schedules, indexed $0, \ldots, 3$, from top to bottom. T illustrates a worst-case interruption as in the proof, which occurs, say, right before contract $x_{3,2}$ terminates. The largest contract completed by time T among all contracts in $\mathcal{X}$ is the third contract of schedule 1, namely $x_{3,1}$.

Theorem 10. *For every $r \geq 4$, define $d = b_r$, if $r \leq \frac{(1+2^n)^2}{2^n}$, and $d = 1 + 2^n$, otherwise. Then* Ideal *has performance $(r, d^{1+\frac{1}{2^n}}/(d-1))$.*

Proof. The worst-case interruptions occur infinitesimally earlier than the completion time of a contract in $\mathcal{X}$. Specifically, let $T = \sum_{i=1}^{j} x_{i,l} - \epsilon$ be an interruption right before the

1. The schedules we define involve modulo arithmetic; for convenience, we thus start their indexing from 0.

j-th contract in X_l completes. We have

$$\sum_{i=1}^{j} x_{i,l} = \sum_{i=1}^{j} d^{i+\frac{l}{2^n}} \leq d^{\frac{l}{2^n}} \frac{d^{j+1}}{d-1}.$$

We consider two cases. If $l \neq 0$, then the largest contract completed by time T in Ideal is $x_{j,l-1}$ of length $d^{j+\frac{l-1}{2^n}}$. The prediction P chooses this schedule, and the consistency is at most $T/x_{j,l-1} \leq \frac{d^{1+\frac{1}{2^n}}}{d-1}$. If $l = 0$, then the largest such contract is contract $x_{j-1,2^n-1}$ of schedule X_{2^n-1} which has length $d^{j-1+\frac{2^n-1}{2^n}} = d^{j-\frac{1}{2^n}}$. Again, the prediction chooses this schedule, and the consistency is at most $T/x_{j-1,2^n-1} \leq \frac{d^{1+\frac{1}{2^n}}}{d-1}$.

Moreover, we require that each schedule in $\mathcal{X}$ is r-robust, or equivalently, that $\frac{d^2}{d-1} \leq r$, as follows from the robustness of exponential schedules (see Section 2).

Thus the best value of d is such that

$$\frac{d^2}{d-1} \leq r \text{ and } \frac{d^{1+\frac{1}{2^n}}}{d-1} \text{ is minimized.}$$

Using standard calculus, it follows that the optimal choice of d is as in the statement of the theorem. □

For $r = 4$, in particular, Theorem 10 shows that Ideal has performance $(2^{1+\frac{1}{2^n}}, 4)$, which matches Theorem 8, and is, therefore, Pareto-optimal.

4.2 Contract scheduling with erroneous queries

Ideal, as its name suggests, is not a practical schedule: a single error in one of the queries can make its acceleration ratio as bad as its robustness. Intuitively, this occurs because the n queries implement a type of "binary search" in the space of all 2^n schedules in $\mathcal{X}$, which is not robust to errors. We will instead propose a family of schedules, which we call $Robust_p$, where $p \in [0, 1]$ is a parameter that defines the range of error that the schedule can tolerate. More precisely, we will define a class of schedules $\mathcal{X}$, and the prediction P will be the index of one of these schedules. However, this time there are only n schedules in $\mathcal{X}$ instead of 2^n, as was in the case of Ideal. Specifically, each $X_i \in \mathcal{X}$ is defined as a near-exponential schedule of the form $X_i = (x_{j,i})_{j\geq 0} = d^{j+\frac{i}{n}}$, with $i \in [0, n-1]$, where the base $d > 1$ will again be determined later.

We now describe the n queries that comprise the prediction P. Each query Q_i, for $i \in [0, n-1]$ is of the form "Is the best schedule, for the given interruption in $\{X_0, \ldots, X_i\}$?". Note that the queries obey a monotonicity property: if Q_i is "yes", and Q_{i+1} is "no", we know that an error has occurred in one of these two queries. Conversely, if Q_i is "no", and Q_{i+1} is "yes", then we know that if all query responses are error-free, then the best schedule is X_{i+1}.

At first glance, one may reasonably think that these queries are overly powerful and must be answered by a powerful oracle. However, as we discuss in Section 4.3, each of the queries Q_i has an equivalent statement as a subset query. Namely, each query asks whether T falls in a certain partition of the timeline, which has a more natural, and practical interpretation.

If there were no errors (i.e., for $\eta = 0$), then the best schedule in $\mathcal{X}$ would be the number of "no" responses to the n queries. However, in the presence of errors, one needs to be careful, because, once again, a single error can have an enormous impact. For this reason, *Robust*$_p$ uses the parameter p. In particular, it chooses schedule X_m, where m is defined as $(N - pn) \bmod n$ and N is the number of "no" responses (again, for convenience we will assume that pn is integral). In words, *Robust*$_p$ chooses a schedule of index "close but less", in the cyclic order of indices, to an index that would correspond to an error-free prediction. The following theorem bounds the performance of *Robust*$_p$, and shows how to choose the base d. We make two assumptions: that $\eta \leq p$ (thus *Robust*$_p$ can only tolerate up to p fraction of query errors), and that $p \leq 1/2$ (otherwise, in the worst case, the query responses are too noisy to be of any use).

Theorem 11. *For every $r \geq 4$, define K to be equal to $\frac{n}{2pn+1}$, and d to be equal to b_r, if $r \leq (1+K)^2/K$, and $1+K$, otherwise. Then Robust$_p$ is r-robust and has acceleration ratio at most $\frac{d^{1+\frac{1}{n}+2p}}{d-1}$, assuming $\eta \leq p \leq 1/2$.*

Proof. For a given interruption T, let l denote the index of the best schedule in $\mathcal{X}$. From the structure of $\mathcal{X}$, this means that, in the worst-case, T occurs right before the completion of a contract, say j, in the schedule $X_{(l+1) \bmod n}$. We will consider the case $l \neq n-1$, thus $(l+1) \bmod n = l+1$ (we will discuss the outlier case $l = n-1$ at the end of the proof). We express this interruption as

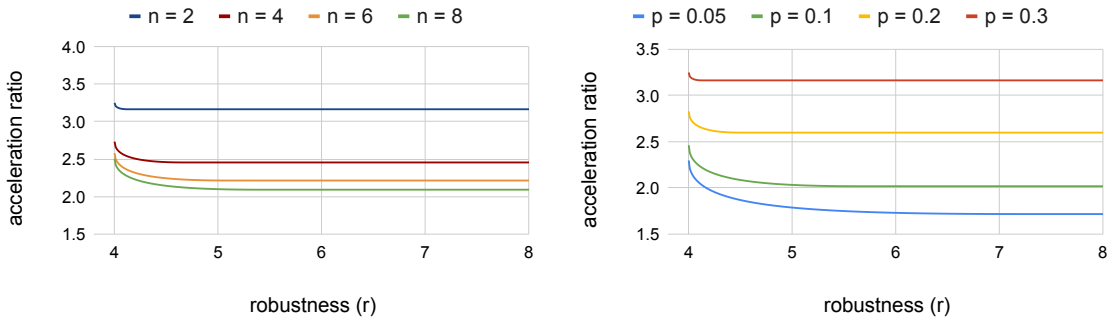
$$T = \sum_{i=1}^{j} x_{i,l+1} = \sum_{i=1}^{j} d^{i+\frac{l+1}{n}} \leq \frac{d^{j+1+\frac{l+1}{n}}}{d-1}.$$

Let m denote the index chosen by *Robust*$_p$, as defined earlier. The crucial observation is that in a cyclic ordering of the indices, m and l are within a distance at most $(\eta + p)n$. Here, a distance of at most ηn is due to the maximum number of erroneous queries, and an additional distance of at most pn is further incurred by the algorithm. Since $\eta \leq p$, they are within a distance at most $2pn$.

We will give a lower bound on the largest contract length, say L completed by time T in *Robust*$_p$. We consider two cases. First, suppose that $m \leq l$, then by the structure of $\mathcal{X}$, L is at least the length $x_{j,l-2pn} = d^{j+\frac{l-2pn}{n}}$. Next, suppose that $m > l$. In this case, L is at least the length of $x_{j-1,n+l-2pn} = d^{j-1+\frac{n+l-2pn}{n}} = d^{j+\frac{l-2p}{n}}$. In both cases we conclude that $L \geq d^{j+\frac{l-2pn}{n}}$. Therefore the acceleration ratio is at most $T/L \leq \frac{d^{1+\frac{1}{n}+2p}}{d-1}$. We now want to find d such that $d^2/(d-1) \leq r$ and $\frac{d^{1+\frac{1}{n}+2p}}{d-1}$ is minimized. Using standard calculus, it follows that the best choice of d is as in the statement of the theorem.

Remains to address the outlier case $l = n-1$. In this case, the proof follows along the same lines, but with a slightly different argument; namely, the worst-case interruption occurs right before the completion time of contract $j+1$ of X_0. Specifically, we have that $T = \frac{d^{j+1}}{d-1}$, and L is at least the length of $x_{j-1,n-1-2pn} = d^{j-1+\frac{n-1-2pn}{n}} = d^{j-\frac{1}{n}-2p}$, and we reach the same conclusion as in the proof shown for the main case. □

For example, consider the case $r = 4$. Then Theorem 11 shows that *Robust*$_p$ is 4-robust and has acceleration ratio at most $2^{1+\frac{1}{n}+2p}$, assuming that $\eta \leq p \leq 1/2$. Figure 6 illustrates the result shown in Theorem 11.



(a) The value of p is fixed to $p = 0.1$.

(b) The number of queries is fixed to $n = 10$

Figure 6: The acceleration ratio of $Robust_p$, as proven in Theorem 11.

4.3 Details on the query implementation

In the statements of Ideal and $Robust_p$, we defined the queries to be of the form "Is the best schedule for interruption T in some subset of $\mathcal{X}$?". One reasonable criticism is that such queries may not be suitable for a practical implementation, since their response requires a very powerful oracle. We explain how one can transform these types of queries to queries of the form "Is T with a certain subset $\mathcal{T}$ of the timeline?", and thus make them more friendly from a practical standpoint. The key idea is to exploit the structure of $\mathcal{X}$, which consists of near-exponential schedules (see also Figure 5).

We begin with Ideal. Let P denote its n-bit prediction, and let $\mathcal{X}_{i,0}$ denote the set of schedules $X_j \in \mathcal{X}$ such that in the binary representation of j, the i-th bit is equal to 0; $\mathcal{X}_{i,1}$ is defined similarly. Then we can think of the i-th bit of P as the response to the query $Q_i =$"Is it preferable to choose a schedule from $\mathcal{X}_{i,0}$ or from $\mathcal{X}_{i,1}$?". Furthermore, this query has an equivalent interpretation in terms of the interruption. In particular, let $\mathcal{T}_{i,0}$ be such that if $T \in \mathcal{T}_{i,0}$, then it is preferable to choose a schedule from $\mathcal{X}_{i,0}$, and similarly for $\mathcal{T}_{i,1}$. Then Q_i is equivalent to "Is T in $\mathcal{T}_{i,0}$ or in $\mathcal{T}_{i,1}$?".

We will argue that it is possible to define succinctly $\mathcal{T}_{i,0}$ and $\mathcal{T}_{i,1}$, by exploiting the structure of $\mathcal{X}$. The main observation is as follows: Suppose that we order the finish times of all contracts in all schedules in $\mathcal{X}$, in increasing order. Let t, t' denote two consecutive finish times, and suppose that t is the finish time of contract x. Then, by construction, $\mathcal{X}$ has the property that contract x is the largest contract completed among all schedules in $\mathcal{X}$, if the interruption occurs in the interval $[t, t')$. Thus, the schedule at which x belongs is, likewise, the best schedule for such an interruption. In particular, consider schedule X_k in $\mathcal{X}$. Then from the above observation, it follows that X_k is the best schedule if $T \in \cup_{j\geq 1}[\sum_{l=1}^{j} x_{l,k}, \sum_{l=1}^{j} x_{l,k+1})$, for the case $k \neq 2^n - 1$, and $T \in \cup_{j\geq 1}[\sum_{l=1}^{j} x_{l,2^k-1}, \sum_{l=1}^{j+1} x_{l,0})$, if $k = 2^n - 1$.

Substituting the $x_{j,i}$ with the corresponding lengths gives a subset of the timeline for which each $X_k \in \mathcal{X}$ is the best choice. Let S_k denote this subset. Then we can define $\mathcal{T}_{i,0}$ as

$$\cup_j \{S_j : \text{the } i\text{-th bit in the binary representation of } j \text{ is } 0\},$$

and similarly $\mathcal{T}_{i,1}$.

Given the above discussion, a similar, and actually simpler interpretation of the queries can be made concerning $Robust_p$. Namely, we can interpret Q_i as "is the interruption T in the set $\cup_{j=0}^{i} S_j$?".

5. Experimental results

In this section, we present the experimental evaluation of our schedules[2]. We use exponential schedules (without any prediction) as the baseline for our comparisons. Specifically, as discussed in Section 2, we know that for any given $r \geq 4$, any exponential schedule $(a^i)_{i\geq 1}$ with base $a \in [c_r, b_r]$ has robustness at most r. For the special, but important case of $r = 4$, there is only one such schedule with base $a = 2$. In our experiments, we report the *empirical acceleration ratio*, namely the ratio $t/\ell(X, t)$, for all t, where X is the evaluated schedule. More precisely, for each discrete value of t (as discussed in more detail later in the section), we generate 1,000 predictions associated with t, in accordance with the two models we study. Each such prediction will have some random noise that simulates prediction error, and we report as $\ell(X, t)$ the average, among the 1,000 predictions, of the length of the largest contract completed by time t in X. Note that similarly to the theoretical worst-case acceleration ratio, it is desirable for a good schedule to achieve small values in terms of its empirical acceleration ratio.

5.1 Interruption time as prediction

We model $\tau \in [T - H, T + H]$ to be a random, normal variable with mean T and standard deviation 1, such that $\eta \leq H$. Recall that an H-aware schedule knows H, whereas an H-oblivious one does not. Figure 7 depicts the average acceleration ratio (y-axis) of the schedule $X^*_{\tau(1-p)}$, as defined in Section 3, for $r = 4$ and for different values of the parameter p, as a function of the interruption time T (x-axis), for fixed $H = 0.1$. The plot depicts the performance of four schedules: the H-aware schedule, in which $p = H = 0.1$, and three H-oblivious schedules for $p = 0.05$, $p = 0.2$ and $p = 0.3$. We run the experiment over 1,000 evenly spaced values of the interruption time in the interval $[2, 2^{20}]$. For each value of the interruption time, the expectation is taken over 1,000 random values of the error.

The plot shows that the H-aware schedule ($p = 0.1$) has an advantage over the schedules with different values of p. In particular, the expected value of the acceleration ratio of this schedule is around 2.23 for all values of the interruption T, compared to acceleration ratios of 2.41 for the schedule with buffer smaller than H ($p = 0.05$), and ratios 2.49 and 2.85 for the schedules whose buffer is larger than H ($p = 0.2, 0.3$, respectively). This is consistent with the analysis in Lemma 4. We also note that the schedule with buffer $p < H$ (i.e., $p = 0.05$) performs quite well, even though it does not obey the conditions of Lemma 4. This is because this schedule performs much better than all other schedules for small negative error, or any positive error, but performs worse for large negative errors. This fact also explains why this particular schedule exhibits the most noisy behavior.

We observe that the acceleration ratio of $X^*_{\tau(1-p)}$ is roughly constant (barring the noise), and independent of T. This is in accordance with the statement of $X^*_{\tau(1-p)}$, since its

2. The code on which the experiments are based is available online at `https://github.com/shahink84/ContractSchdulingWithPredictions`.

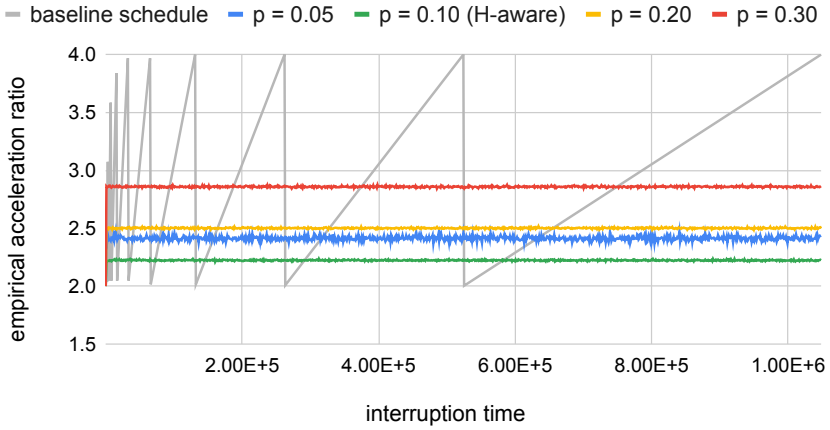


Figure 7: Acceleration ratios of $X^*_{\tau(1-p)}$, for $r = 4$ and $H = 0.1$.

acceleration ratio is determined by its largest contract, i.e., a contract of size $\frac{T(1-p)}{c_r}$, as discussed in Section 3. As p decreases, the fluctuation of the acceleration ratio due to the random error increases, since the interruption becomes closer to the completion time of its largest completed contract by time $\tau(1-p)$.

As Figure 7 shows, our schedules with predictions do not outperform the baseline algorithm for every interruption. This is to be expected, since there is no schedule that can dominate any other schedule. More precisely, even a schedule of very bad robustness (e.g., a schedule with a huge contract early on) will have excellent acceleration ratio for some range of interruptions (e.g., for certain interruptions before the completion time of the huge contract). Nevertheless, we can quantify the advantage of the schedules with predictions, as shown in Table 1. The table depicts the percentage of interruptions in $[2, 2^{20}]$ for which $X^*_{\tau(1-p)}$ outperforms the baseline schedule, as well as the percentage of interruptions for which the improvement is significant (at least by 20%). As expected, the H-aware schedule yields the best improvements, but even the H-oblivious schedules tend to perform much better than the baseline schedule. The conclusion is that while H-awareness yields a clear improvement, it is not indispensable.

	$p = 0.05$	$p = 0.1$	$p = 0.2$	$p = 0.3$
improvement	79.22%	88.71%	74.73%	57.04%
strong improvement	55.24%	66.43%	50.05%	28.47%

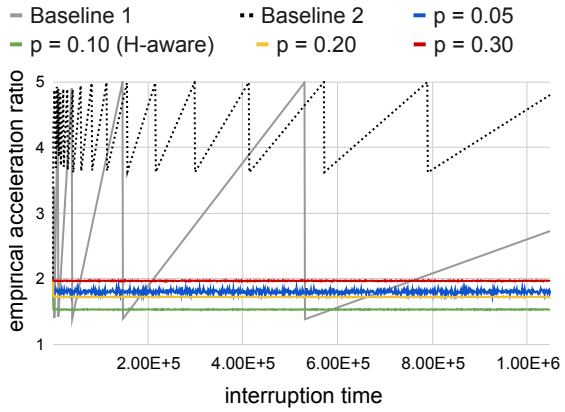
Table 1: Percentage of interruptions in $[2, 2^{20}]$ for which $X^*_{\tau(1-p)}$ outperforms the baseline schedule, for the setting of Figure 7.

5.1.1 Experiments on the robustness r

We evaluate and compare the performance of schedules for $r \in \{5, 7\}$, $H = 0.1$, and prediction error that is generated as described earlier in the section. For each r we consider two exponential schedules (without predictions) as baseline schedules, namely one with base

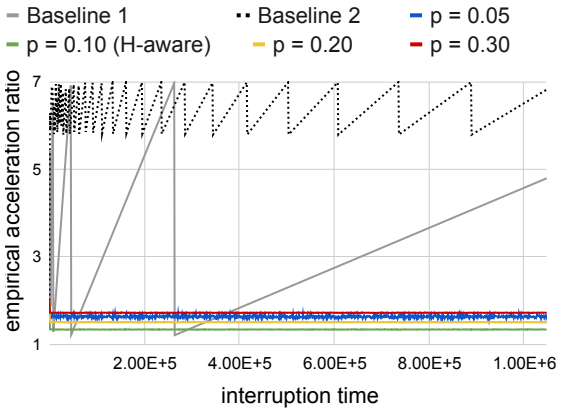
b_r (Baseline 1) and one with base c_r (Baseline 2). We observe that, although incomparable to each other, Baseline 1 tends to have smaller empirical acceleration ratio than Baseline 2 for relatively larger time intervals, and in this sense may be preferable. For this reason, we report the comparison of our schedules to Baseline 1.

Figure 8 depicts the performance of the different schedules. The associated tables show the percentage of interruptions for which the schedule with prediction and buffer p outperforms Baseline 1, as well as the percentage of interruptions for which this improvement is significant (by at least 20%). We observe that as r increases, the improvements of the schedules with predictions become more pronounced. This is consistent with Lemma 4, since c_r is decreasing function of r.



	$p = 0.05$	$p = 0.1$	$p = 0.2$	$p = 0.3$
improv.	78.22%	92.50%	82.51%	70.02%
strong improv.	60.63%	76.52%	65.33%	50.14%

(a) $r = 5$



	$p = 0.05$	$p = 0.1$	$p = 0.2$	$p = 0.3$
improv.	89.51%	96.60%	92.40%	86.91%
strong improv.	80.61%	90.00%	84.81%	78.42%

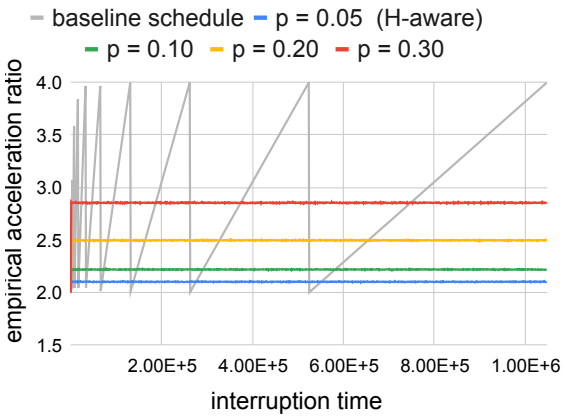
(b) $r = 7$

Figure 8: Plots of the acceleration ratios as function of T, for the baseline schedules, and schedule $X^*_{\tau(1-p)}$, for $r \in \{5, 7\}$ and $H = 0.1$. The tables show the percentage of interruptions for which $X^*_{\tau(1-p)}$ outperforms, and strongly outperforms Baseline 1.

5.1.2 Experiments on the error

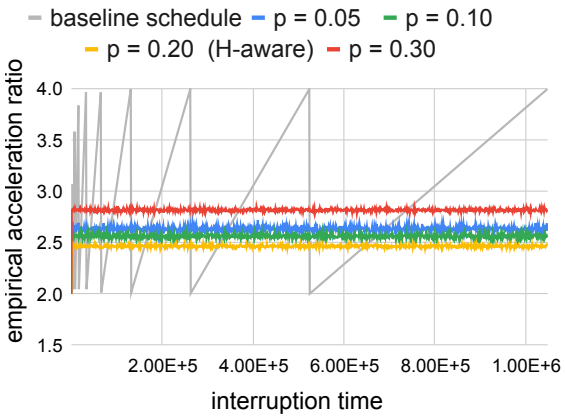
We report experiments on the performance of the schedules as function of different values of error. First, Figure 9 depicts the acceleration ratio of the schedules for $H \in \{0.05, 0.2\}$ and $r = 4$. The error is again modeled as described earlier, namely the prediction τ is equal to $T/(1 + x)$, where x is a normal random variable in the range $[-H, H]$, with standard deviation equal to 1. As before, the associated tables show the percentage of interruptions for which the schedule with prediction and buffer p outperforms the baseline schedule, as well as the percentage of interruptions for which this improvement is significant (by at least 20%).

The predictions remain consistently beneficial, even for relatively large bound on error $H = 0.2$. As expected, the advantage of schedules with prediction decreases as H increases. For example, for $p = 0.1$, the improvement over the baseline schedule decreases from about 89% to about 71% when H is increased from 0.05 to 0.2. Once again, the H-aware schedules are the best-performing. We observe that the plots become more noisy as H increases. This is again to be expected, since the magnitude of the error increases. Furthermore, as H increases, so are the chances that the different schedules will exhibit similar behavior, relative to the sign and the magnitude of the error, which explains why they are much closer, in terms of performance, for large H.



	$p = 0.05$	$p = 0.1$	$p = 0.2$	$p = 0.3$
improv.	94.50%	88.81%	74.92%	57.04%
strong improv.	73.82%	66.63%	50.14%	28.57%

(a) $H = 0.05$

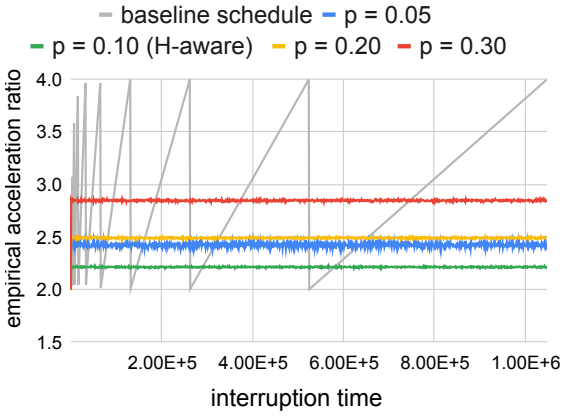


	$p = 0.05$	$p = 0.1$	$p = 0.2$	$p = 0.3$
improv.	68.13%	71.82%	76.82%	59.24%
strong improv.	41.65%	46.35%	51.94%	30.86%

(b) $H = 0.2$

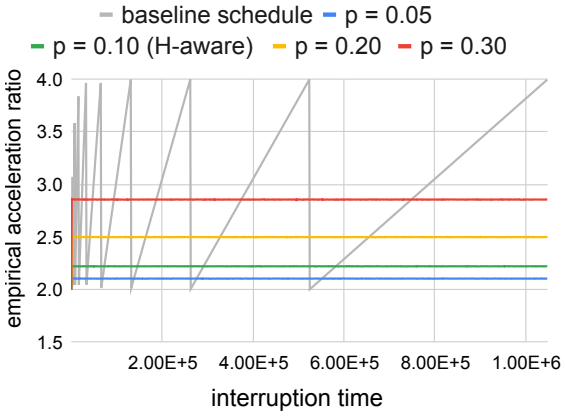
Figure 9: Plots of the acceleration ratios as function of T, for the baseline schedule and schedule $X^*_{\tau(1-p)}$, for $r = 4$ and $H \in \{0.05, 0.2\}$. The tables show the percentage of interruptions for which $X^*_{\tau(1-p)}$ outperforms, and strongly outperforms the baseline schedule.

Next, we consider different distributions for generating the prediction error. Figure 10a depicts the acceleration ratio of different schedules when τ is equal to $T/(1 + x)$, where x is a *uniform* random variable in the range $[-H, H]$, and for the setting $H = 0.1$ and $r = 4$. Figure 10b depicts the acceleration ratios when x is a truncated normal variable with mean 0 and standard deviation 0.01. In both cases, we consider $r = 4$ and $H = 0.1$. When compared to the error model used in our previous experiments, these two distributions capture two extreme values for the standard deviation of the truncated normal variable x (when the standard deviation becomes large, a truncated normal variable resembles the uniform distribution). As earlier, the associated tables show the percentage of interruptions for which the schedule with prediction and buffer p outperforms the baseline schedule, as well as the percentage of interruptions for which this improvement is significant (by at least 20%).



	$p = 0.05$	$p = 0.1$	$p = 0.2$	$p = 0.3$
improv.	78.42 %	88.91%	75.52%	57.54%
strong improv.	54.54%	66.73%	50.34%	29.07%

(a) uniform distribution



	$p = 0.05$	$p = 0.1$	$p = 0.2$	$p = 0.3$
improv.	94.50%	88.81%	74.82%	57.04%
strong improv.	73.82%	66.63%	50.04%	28.57%

(b) truncated normal distribution with standard deviation 0.01

Figure 10: Plots of the acceleration ratios as function of T, for the baseline schedule, and schedule $X^*_{\tau(1-p)}$, for $r = 4$ and $H = 0.1$ when the prediction error follows (a) uniform distribution and (b) a truncated normal distribution with a small standard deviation equal to 0.01. The tables show the percentage of interruptions for which $X^*_{\tau(1-p)}$ outperforms, and strongly outperforms the baseline schedule.

We observe similar outcomes as in the earlier experiments, in regards to the relative performance of the various schedules. Figure 10b shows that the schedule with buffer $p = 0.05$ has the best performance. This is because the standard deviation is now small, and the prediction is more concentrated around the mean. A smaller buffer p can be beneficial in this case, even if it is smaller than H, since the negative error will tend to be small with high probability. Schedules with $p > H$ remain essentially unaffected, since they are tailored to worst-case analysis (i.e., consider the error to be as high as H).

5.2 Query-based predictions

We evaluate experimentally the performance of the schedule $Robust_p$ (as explained in Section 4.1, Ideal is only a theoretical schedule). We fix the number n of queries to be equal to 100, and we set $H = 0.1$. Given a perfect binary prediction of size 100, i.e., perfect responses to 100 queries associated with $Robust_p$, we generate a noisy prediction by flipping a fraction η of the 100 bits (rounded down) where η is chosen uniformly at random in $[0, H]$. Figure 11 depicts the average acceleration ratio (y-axis) of $Robust_p$ for different values of the parameter p, as a function of the interruption time T (x-axis). As in the experiments in Section 5.1, the expectation is taken over 1,000 random values of the error, and the interruption time takes values in the interval $[2, 2^{20}]$.

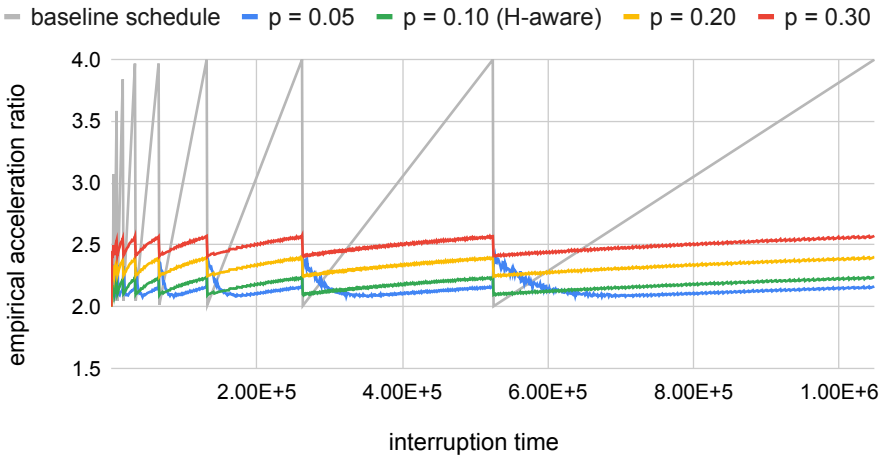


Figure 11: Acceleration ratios of $Robust_p$, for $H = 0.1$.

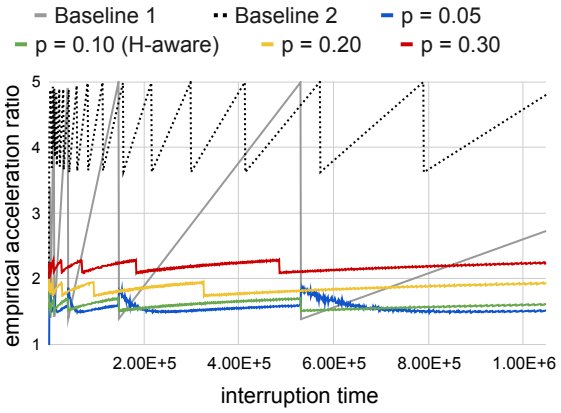
We evaluate $Robust_p$ with four values of the parameter p, namely $p \in \{0.05, 0.1, 0, 2, 0.3\}$. Note that the theoretical upper bound of Theorem 11 applies only if $p \geq H = 0.1$ in this setting. For such values of p, the acceleration ratio is a "saw-like" function of the interruption. Namely, there are some "critical" interruptions at which the empirical acceleration ratio drops, then increases until the next critical interruption, and the pattern repeats. These critical interruptions are the points in time at which a better contract can be chosen in the collection of schedules $\mathcal{X}$ on which $Robust_p$ is based, i.e., when the parameter l in the proof of Theorem 11 increases by one. In between critical interruptions, the acceleration ratio increases, since the "best" contract does not change. The acceleration ratio of $Robust_p$ increases with p, as predicted by Theorem 11, but is much smaller than the baseline acceleration ratio; for instance, for $p = 0.3$, it fluctuates in the interval $[2.4, 2.6]$.

Note that even for $p = 0.05 < H$, $Robust_p$ performs better than the baseline schedule, which is interesting because such a case is not captured by the worst-case analysis in Theorem 11. This implies that $Robust_p$ may work in practice for a wider range of values of p than predicted by the theorem, and that $Robust_p$ need not be H-aware to perform well. However, such a schedule is more sensitive to error. This is because for some interruptions, it will complete a rather inefficient contract of small length, i.e., one that corresponds to a schedule of index close but *larger* than l in the cyclic order (where l refers to the index of the best schedule in the proof of Theorem 11); for some other interruptions, however, it may end up completing a large contract (one that corresponds to a schedule of index close but smaller than l in the cyclic order).

In Table 2 we report the performance gain of $Robust_p$ for different values of p, and $H = 0.1$. Once again, the table shows the percentage of interruptions in the range $[2, 2^{20}]$ for which $Robust_p$ outperforms the baseline schedule, as well as the percentage of interruptions for which the performance gain is significant (at least 20%).

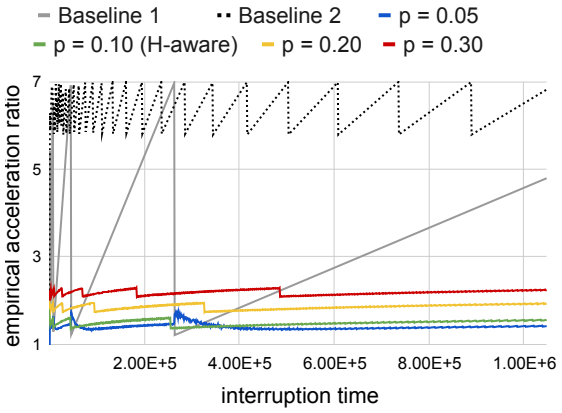
	$p = 0.05$	$p = 0.1$	$p = 0.2$	$p = 0.3$
improvement	89.81%	94.25%	86.07%	77.07%
strong improvement	74.33%	70.98%	60.94%	49.95%

Table 2: Percentage of interruptions in $[2, 2^{20}]$ for which $Robust_p$ outperforms the baseline schedule, for the setting of Figure 11.



	$p = 0.05$	$p = 0.1$	$p = 0.2$	$p = 0.3$
improv.	78.22%	92.50%	82.51%	70.02%
strong improv.	60.63%	76.52%	65.33%	50.14%

(b) $r = 5$



	$p = 0.05$	$p = 0.1$	$p = 0.2$	$p = 0.3$
improve.	93.40%	95.30%	85.41%	74.72%
strong improv.	88.21%	87.81%	75.32%	66.03%

(d) $r = 7$

Figure 12: Plots of the acceleration ratios as function of T, for the baseline schedules, and schedule $X^*_{\tau(1-p)}$, for $r \in \{5, 7\}$ and $H = 0.1$. The tables show the percentage of interruptions for which $Robust_p$ outperforms, and strongly outperforms Baseline 1.

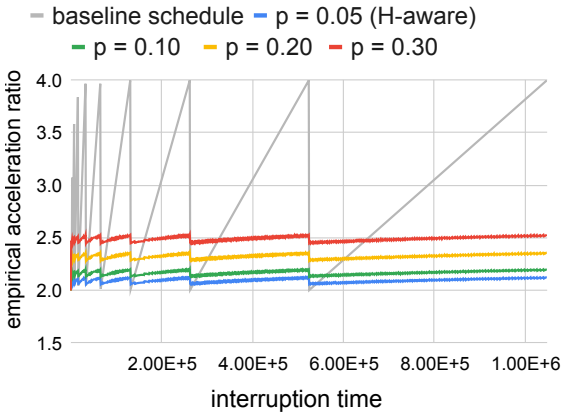
5.2.1 Experiments on the robustness

We report experimental results on $Robust_p$ for other values of the robustness parameter, namely for $r \in \{5, 7\}$, and $H = 0.1$. As discussed in Section 5.1.1, we consider two baseline, exponential schedules without predictions, one for which the base is as large as possible (Baseline 1, with base equal to b_r), and one for which it is as small as possible (Baseline 2, with base equal to c_r). We consider error generated uniformly at random in $[0, H]$. Figure 12 illustrates the results, and the associated tables show the percentage of interruptions for which $Robust_p$ outperforms Baseline 1.

We observe the same relative ordering of the different schedules in $Robust_p$, as for the case $r = 4$. Moreover, as the robustness guarantee r increases, the schedules with predictions have more leeway for improving their consistency and the acceleration ratio, which is reflected in the attained ratios, and the improvement in comparison to the baseline schedule.

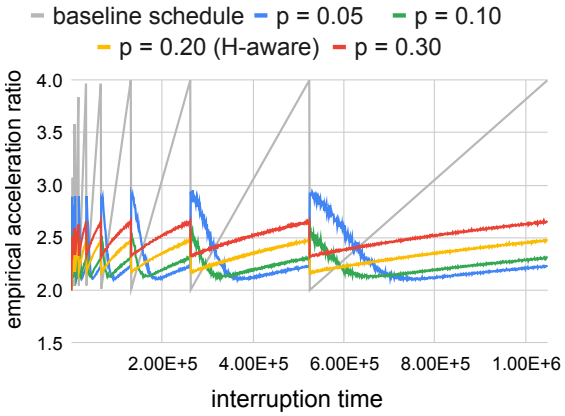
5.2.2 Experiments on the error

We report experimental results on $Robust_p$ for other values of the parameter H, assuming $r = 4$, and that the error is generated uniformly at random. Figure 13 depicts the acceleration ratio of the different schedules when $H \in \{0.05, 0.2\}$.



	$p = 0.05$	$p = 0.1$	$p = 0.2$	$p = 0.3$
improv.	96.20%	92.40%	85.11%	75.82%
strong improv.	74.92%	70.52%	60.53%	50.64%

(a) $H = 0.05$



	$p = 0.05$	$p = 0.1$	$p = 0.2$	$p = 0.3$
improv.	79.92%	87.31%	89.01%	79.32%
strong improv.	69.23%	71.72%	61.43%	48.85%

(b) $H = 0.2$

Figure 13: Plots of the acceleration ratios as function of T, for the baseline schedule, and schedule $Robust_p$, for $r = 4$ and $H \in 0.05, 0.2$. The tables show the percentage of interruptions for which $Robust_p$ outperforms, and strongly outperforms the baseline schedule.

As the figures and tables illustrate, $Robust_p$ is consistently better than the baseline schedule. As p increases, and as long as $p \geq H$ (hence Theorem 11 applies), the performance of $Robust_p$ is monotone with the error. For $p < H$, $Robust_p$ will still perform better than the baseline schedule, but the worst-case empirical acceleration ratio is markedly worse than schedules for $p \geq H$, as we explained earlier in this section. This finding shows that the assumption that $p \leq H$ is in a sense requisite in the theoretical analysis of $Robust_p$, in order to establish strict, worst-case analytical guarantees.

6. Conclusion

In this work we studied a classic problem from the domain of bounded-resource reasoning, namely the design of interruptible algorithms based on contract scheduling, in a setting in which there is some prediction concerning the time at which the interruptible system will be queried. We studied two prediction models, both from the point of view of theoretical and experimental analysis. The first model is motivated by learning-augmented algorithms, and considers the interruption time explicitly as the prediction. The second model is a novel framework for eliciting predictions by means of responses to binary queries. We explored

tradeoffs between the prediction accuracy, the acceleration ratio, the consistency and the robustness of schedules.

It is intriguing that although contract scheduling has a relatively simple statement and solution in the standard, prediction-free setting, the problem becomes far more challenging under the predictions framework. In future work, we would like to study extensions of contract scheduling, such as scheduling contract algorithms for multiple instances, in a single or multiple parallel processors, which have been studied extensively in the prediction-free setting, as discussed in Section 1.

Another direction is to investigate connections between contract scheduling and *online searching under the competitive ratio* with untrusted advice. Recent work (Angelopoulos, 2021) studied this problem strictly from the point of view of consistency/robustness tradeoffs. The techniques we developed and the results we showed in this work could be applicable in searching with noisy, erroneous advice, given the known connections between contract scheduling and searching under the competitive ratio (Bernstein et al., 2003; Angelopoulos, 2015).

It is known that randomization can help improve the performance of contract scheduling, and the best randomized acceleration ratio is e (Chrobak & Kenyon-Mathieu, 2006) (as opposed to the best deterministic acceleration ratio that is equal to 4). It would be interesting to study the effect of randomization in the consistency/robustness tradeoffs, especially since many of the proofs in this work apply to the deterministic setting.

The query-based prediction model we introduced can apply naturally to many other optimization problems. For instance, an interesting direction is to study *clustering* with noisy queries. Specifically, unlike the setting described in (Mazumdar & Saha, 2017), one would aim to establish performance guarantee tradeoffs without any probabilistic assumptions on the query responses. Last, we would like to explore connections between the performance under the query-based model and the very rich field of *fault-tolerant search* (Pelc, 2002; Cicalese, 2013), with the aim to improve the choice of a good algorithm from a set of candidate algorithms (which is the main conceptual idea in the proof of Theorem 11).

Acknowledgements

This research was supported by the CNRS-Emergence project ONFIN, and by the project PREDICTIONS, grant ANR-19-CE48-0016 from the French National Research Agency (ANR). We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) [funding reference number DGECR-2018-00059].

References

Angelopoulos, S. (2015). Further connections between contract-scheduling and ray-searching problems. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1516–1522.

Angelopoulos, S. (2021). Online search with a hint. In *Proceedings of the 12th Innovations in Theoretical Computer Science Conference (ITCS)*, pp. 51:1–51:16.

Angelopoulos, S., Dürr, C., Jin, S., Kamali, S., & Renault, M. P. (2020). Online computation with untrusted advice. In *Proceedings of the 11th International Conference on Innovations in Theoretical Computer Science (ITCS)*, pp. 52:1–52:15.

Angelopoulos, S., & Jin, S. (2019). Earliest-completion scheduling of contract algorithms with end guarantees. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence, (IJCAI)*, pp. 5493–5499.

Angelopoulos, S., & Kamali, S. (2021). Contract scheduling with predictions. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence, AAAI 2021 2021*, pp. 11726–11733. AAAI Press.

Angelopoulos, S., Kamali, S., & Zhang, D. (2022). Online search with best-price and query-based predictions. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence*, pp. 9652–9660. AAAI Press.

Angelopoulos, S., & López-Ortiz, A. (2009). Interruptible algorithms for multi-problem solving. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 380–386.

Angelopoulos, S., López-Ortiz, A., & Hamel, A. (2008). Optimal scheduling of contract algorithms with soft deadlines. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI)*, pp. 868–873.

Antoniadis, A., Coester, C., Elias, M., Polak, A., & Simon, B. (2020). Online metric algorithms with untrusted predictions. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pp. 11453–11463.

Bernstein, D. S., Finkelstein, L., & Zilberstein, S. (2003). Contract algorithms and robots on rays: Unifying two scheduling problems. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1211–1217.

Bernstein, D. S., Perkins, T. J., Zilberstein, S., & Finkelstein, L. (2002). Scheduling contract algorithms on multiple processors. In *Proceedings of the 18th AAAI Conference on Artificial Intelligence (AAAI)*, pp. 702–706.

Boddy, M., & Dean, T. L. (1994). Deliberation scheduling for problem solving in time-constrained environments. *Artif. Intell.*, *67*(2), 245–285.

Chrobak, M., & Kenyon-Mathieu, C. (2006). SIGACT news online algorithms column 10: Competitiveness via doubling. *SIGACT News*, *37*(4), 115–126.

Cicalese, F. (2013). *Fault-Tolerant Search Algorithms - Reliable Computation with Unreliable Information*. Monographs in Theoretical Computer Science. An EATCS Series. Springer.

Eberle, F., Lindermayr, A., Megow, N., Nölke, L., & Schlöter, J. (2022). Robustification of online graph exploration methods. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence, AAAI 2022*, pp. 9732–9740. AAAI Press.

Gollapudi, S., & Panigrahi, D. (2019). Online algorithms for rent-or-buy with expert advice. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pp. 2319–2327.

Horvitz, E. (1988). Reasoning about beliefs and actions under computational resource constraints. *Int. J. Approx. Reasoning*, *2*(3), 337–338.

Im, S., Kumar, R., Qaem, M. M., & Purohit, M. (2021). Online knapsack with frequency predictions. In *Proceedings of the 34th Annual Conference on Neural Information Processing Systems (NeurIPS)*, pp. 2733–2743.

Kupavskii, A., & Welzl, E. (2019). Lower bounds for searching robots, some faulty. *Distributed Computing*, *34*(4), 229–237.

Lattanzi, S., Lavastida, T., Moseley, B., & Vassilvitskii, S. (2020). Online scheduling via learned weights. In *Proceedings of the 30th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 1859–1877.

Lee, R., Maghakian, J., Hajiesmaili, M. H., Li, J., Sitaraman, R. K., & Liu, Z. (2021). Online peak-aware energy scheduling with untrusted advice. In *e-Energy '21: The Twelfth ACM International Conference on Future Energy Systems*, pp. 107–123. ACM.

Li, T., Yang, R., Qu, G., Shi, G., Yu, C., Wierman, A., & Low, S. H. (2022). Robustness and consistency in linear quadratic control with untrusted predictions. *Proc. ACM Meas. Anal. Comput. Syst.*, *6*(1), 18:1–18:35.

López-Ortiz, A., Angelopoulos, S., & Hamel, A. (2014). Optimal scheduling of contract algorithms for anytime problem-solving. *J. Artif. Intell. Res.*, pp. 533–554.

Lykouris, T., & Vassilvitskii, S. (2018). Competitive caching with machine learned advice. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pp. 3302–3311.

Mazumdar, A., & Saha, B. (2017). Clustering with noisy queries. In *Annual Conference on Neural Information Processing Systems (NIPS)*, Vol. 30, pp. 5788–5799.

Mitzenmacher, M. (2020). Scheduling with predictions and the price of misprediction. In *Proceedings of the 11th Innovations in Theoretical Computer Science Conference (ITCS)*, Vol. 151, pp. 14:1–14:18.

Mitzenmacher, M., & Vassilvitskii, S. (2020). Algorithms with predictions. In *Beyond the Worst-Case Analysis of Algorithms*, pp. 646–662. Cambridge University Press.

Pelc, A. (2002). Searching games with errors - fifty years of coping with liars. *Theor. Comput. Sci.*, *270*(1-2), 71–109.

Purohit, M., Svitkina, Z., & Kumar, R. (2018). Improving online algorithms via ML predictions. In *Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 9661–9670.

Rivest, R. L., Meyer, A. R., Kleitman, D. J., Winklmann, K., & Spencer, J. (1980). Coping with errors in binary search procedures. *J. Comput. Syst. Sci.*, *20*(3), 396–404.

Russell, S. J., & Zilberstein, S. (1991). Composing real-time systems. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 212–217.

Sun, B., Lee, R., Hajiesmaili, M., Wierman, A., & Tsang, D. (2021). Pareto-optimal learning-augmented algorithms for online conversion problems. In *Proceedings of the 34th Annual Conference on Neural Information Processing Systems (NeurIPS)*, pp. 10339–10350.

Wei, A., & Zhang, F. (2020). Optimal robustness-consistency trade-offs for learning-augmented online algorithms. In *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS)*.

Xu, C., & Moseley, B. (2022). Learning-augmented algorithms for online Steiner trees. In *AAAI*, pp. 8744–8752. AAAI Press.

Zilberstein, S. (1996). Using anytime algorithms in intelligent systems. *AI Magazine*, *17*(3), 73–83.

Zilberstein, S., & Russell, S. J. (1993). Anytime sensing, planning and action: A practical model for robot control. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1402–1407.

Zilberstein, S., Charpillet, F., & Chassaing, P. (2003). Optimal sequencing of contract algorithms.. *Ann. Math. Artif. Intell.*, *39*(1-2), 1–18.

Zilberstein, S., & Russell, S. J. (1996). Optimal composition of real-time systems. *Artif. Intell.*, *82*(1-2), 181–213.

Appendix

Proof of Corollary 3. It is known that for any $r \geq 4$, any $(1+2r)$-competitive algorithm for searching on the line can be translated as a schedule of acceleration ratio r, and vice versa, see e.g. (Chrobak & Kenyon-Mathieu, 2006). Therefore, the same linear recurrence relation $\sum_{j=1}^{i} x_j \leq r x_{i-1}$ can describe the performance of r-robust strategies for both problems. From Corollary 3 and the discussion in Theorem 4 in (Angelopoulos, 2021), it follows that for any fixed $\epsilon' > 0$, there exists some i_0 such that for all $i \geq i_0$ we have that

$$\sum_{j=1}^{i-1} x_j \geq \frac{x_i}{(b_r - 1)(1-\epsilon')}.$$

Therefore,

$$\sum_{j=1}^{i} x_j \geq x_i + \frac{x_i}{(b_r - 1)(1-\epsilon')} \geq x_i \frac{b_r}{b_r - 1}(1-\epsilon), \tag{10}$$

where ϵ is a function of ϵ', and thus can be arbitrarily small as well. Moreover, from the definitions of c_r and b_r given in (3), it readily follows that $c_r = \frac{b_r}{b_r - 1}$, and thus the second part of the corollary follows from (10). For the first part, note that for given schedule X and time t, the ratio $t/\ell(X,t)$ is minimized when t is right after the completion of a contract, say x_i, therefore, from (10) we have that

$$\ell(X,t) \leq x_i \frac{t}{t} \leq x_i \frac{t}{c_r x_i (1-\epsilon)} = \frac{t}{c_r(1-\epsilon)},$$

where the last inequality follows from the first part of the corollary. Last, note that $1/(1-\epsilon)$ can be written equivalently as $1+\epsilon''$, which completes the proof of the first part of the corollary as well. □

Details in the proof of Lemma 9. For the induction hypothesis, suppose that the lemma holds for $l-1$. We will next show that it holds for l.

First, we will prove the lower bound on y_l. Consider an interruption infinitesimally earlier than $T_{\bar{i}_{l-1},l-1}$. From the induction hypothesis, we know that for this interruption, Z must choose a schedule in $\{X_l, \dots X_{l-1}\}$. The largest contract finished by that time is y_l. Thus it must be that

$$\begin{aligned}
\frac{T_{\bar{i}_{l-1},l-1}}{y_l} < S \Rightarrow y_l &\geq \frac{T_{\bar{i}_{l-1},l-1}}{S} \\
&\geq \frac{2y_{l-1}}{y_l} && \text{(From (7))} \\
&\geq 2\frac{2^{1-\frac{l-2}{2^n}}\alpha}{2^{1+\frac{1}{2^n}}} && \text{(From the induction hypothesis)} \\
&= 2^{1-\frac{l-1}{2^n}}\alpha.
\end{aligned}$$

Consider now an interruption infinitesimally earlier than $T_{\bar{i}_l,l}$. Suppose first, by way of contradiction, that Z chooses X_1. Then it must be that

$$\begin{aligned}
\frac{T_{\bar{i}_l,l}}{x_{i-1},1} < S \Rightarrow S &> \frac{2y_l}{\alpha} && \text{(From (7))} \\
&\geq 2\frac{2^{1-\frac{l-1}{2^n}}\alpha}{\alpha} && \text{(Since } y_l \geq 2^{1-\frac{l-1}{2^n}}\alpha\text{)} \\
&\geq 2 \cdot 2^{1-\frac{2^n-1}{2^n}} && \text{(Since } l \leq 2^n\text{)} \\
= 2^{1+\frac{1}{2^n}},
\end{aligned}$$

a contradiction.

Suppose then, again by way of contradiction, that Z chooses one of $X_2, \dots X_{l-1}$, say X_m. We will arrive to a contradiction by applying an argument similar to the one we used for the base case. Namely, it must be that

$$\frac{T_{\bar{i}_l,l}}{z_m} \leq S, \text{ and } \; z_m \leq \frac{T_{\bar{i}_l,l} - y_m}{2},$$

from which we obtain that

$$S \geq 2\frac{T_{\bar{i}_l,l}}{T_{\bar{i}_l,l} - y_m},$$

and using the same argument as in (8) and (9) it follows that $S \geq 2^{2-\frac{1}{2^n}}$, a contradiction.

We conclude that for the above-defined interruption, Z cannot choose a schedule in $\{X_1, \dots X_l\}$. This completes the inductive step, and the proof of the lemma. □