Appendix

A Details from Section 3

Proof of Theorem 3.1. Let A be a Pareto-optimal algorithm of robustness r, and consistency $c(r)$. We will show that for any fixed $\epsilon > 0$, there exists a sequence σ and a prediction $\hat{p}$ such that $\eta = |\hat{p} - p^*_\sigma| \leq \epsilon$, and A satisfies Definition 3.1. Since A is Pareto-optimal, there exists a non-empty set of sequences Σ_c, such that for all $\sigma_c \in \Sigma_c$, if A is given as prediction $p^*_{\sigma_c}$, then

$$\frac{p^*_{\sigma_c}}{A(\sigma_c)} = c(r).$$

As shown in [19] we can assume, without loss of generality, that every σ_c is increasing, i.e., it is of the form $\sigma_c = p_1, \ldots, p_k, p^*_{\sigma_c}$ with $p_i > p_j$, for all $i < j$, and $p^*_{\sigma_c} > p_k$. We define Σ to be the co-domain of the following function, f:

$$f\colon \Sigma_c \to \Sigma \text{ such that } \; f(\sigma_c) = \begin{cases} \sigma_c & \text{if } |p^*_{\sigma_c} - p_k| \leq \epsilon, \\ p_1, \ldots, p_k, p^*_{\sigma_c} - \epsilon, p^*_{\sigma_c} & \text{otherwise.} \end{cases} \tag{A.1}$$

Given a $\sigma \in \Sigma$, let $n = |\sigma| - 1$, and let x_n be the fraction exchanged by A. Since A is r-robust, it needs to account for the scenario in which the adversary chooses to drop all rates to 1 after exchanging at the rate p_n. Thus, x_n must satisfy

$$\frac{p_n}{s_n + p_n \cdot x_n + 1 - x_n - w_n} \leq r,$$

or equivalently,

$$x_n \geq \frac{p_n - r \cdot (s_n + 1 - w_n)}{r \cdot (p_n - 1)}. \tag{A.2}$$

Define ω to be the RHS of (A.2) Suppose first, that there exists a sequence $\sigma \in \Sigma$ for which A exchanges $x_n = \omega$. In this case, if A is given a prediction $\hat{p} = p^*_\sigma$, then for the the sequence $\sigma_r = \sigma[1, n]$ we have that $|\hat{p} - p^*_{\sigma_r}| \leq \epsilon$, and:

$$\frac{p^*_{\sigma_r}}{A(\sigma_r)} = \frac{p_n}{s_n + p_n \cdot \omega + 1 - \omega - w_n} = r,$$

and the proof is complete in this case.

It thus remains to consider the case that for all $\sigma \in \Sigma$, $x_n > \omega$. Let x_{n+1} be the amount exchanged by A at rate p^*_σ. We define an online algorithm A', whose statement is given in Algorithm 3. Intuitively, while the rate is below p^*_σ, A' makes the same decisions as A. If the rate is between $p^*_\sigma - \epsilon$ and p^*_σ, A' exchanges ω. If the rate is precisely p^*_σ A' exchanges x_n plus what A did not exchange on rates which were between $p^*_\sigma - \epsilon$ and p^*_σ. Finally, A' makes the same decisions as A for all rates that exceed p^*_σ. We will show that A' has robustness at most r and consistency $c_{A'}$ such that $c_{A'} < c(r)$, which contradicts that A is Pareto-optimal.

We first show that A' is r-robust. Let σ' be an input sequence and $\hat{p}$ a prediction given to A', we will show that $p^*_{\sigma'} \leq rA(\sigma')$. If $p^*_{\sigma'} < \hat{p} - \epsilon$, then has A' made the same decisions as A, hence remains r-robust. If $\hat{p} - \epsilon < p^*_{\sigma'} < \hat{p}$, then by definition of ω, A' is guaranteed to be r-robust. Last, if $p^*_{\sigma'} \geq \hat{p}$, then A' achieves a strictly better profit than A.

It remains to show that A' has consistency strictly smaller than $c(r)$. To this end, it suffices to show that: (i) for all $\sigma_c \in \Sigma_c$ it holds that $\frac{\textsc{Opt}(\sigma_c)}{A'(\sigma_c)} < c(r)$, and that (ii) for all $\sigma' \notin \Sigma_c$ it holds that $\frac{\textsc{Opt}(\sigma_c)}{A'(\sigma_c)} < c(r)$, assuming that both A and A' are given a prediction $\hat{p} = p^*_{\sigma'}$.

To show (i), note that for $\sigma' \in \Sigma_c$ it holds that $\frac{\textsc{Opt}(f(\sigma'))}{A(f(\sigma'))} < c(r)$, due to A exchanging $x_n > \omega$ and A' exchanging $x_n = \omega$. If $f(\sigma') = \sigma'$ (first case in (A.1)) then $\frac{\textsc{Opt}(\sigma')}{A(\sigma')} < c(r)$. Otherwise, (second case in (A.1)) $A(\sigma') > A(f(\sigma'))$ hence the same result holds. To show (ii), observe that $A'(\sigma') > A(\sigma')$ due to A exchanging $x_n > \omega$ and A' exchanging $x_n = \omega$. Hence, by the definition of Σ_c, we have

Algorithm 3 Statement of the online algorithm A'

Input: Algorithm $A, \hat{p}, \epsilon$

$p^* = 1,\ e \leftarrow 0$
for each rate p_i in the input sequence **do**
  if $p_i > p^*$ **then**
    $p^* \leftarrow p_i$
    if $p_i < \hat{p} - \epsilon$ **then**
      Exchange the same amount as A
    else if $\hat{p} - \epsilon < p_i < \hat{p}$ **then**
      Exchange ω
      $e \leftarrow e + x_i - \omega$
    else if $p_i = \hat{p}$ **then**
      Exchange $x_n + e$
    else
      Exchange the same amount as A

$$\frac{\textsc{Opt}(\sigma_c)}{A'(\sigma_c)} < \frac{\textsc{Opt}(\sigma_c)}{A(\sigma_c)} < c(r),$$

which concludes the proof. □

B Details from Section 4

In this section, we show how to compute the function Φ used in Profile (Algorithm 1), for deciding whether a profile F is feasible. Recall that we seek a function Φ and values $0 = w_1 \leq \ldots \leq w_{l+1} \leq 1$ that satisfy the following sets of constraints.

$$[\beta] \qquad \forall \beta \in [w_i, w_{i+1}) : \frac{\Phi(\beta)}{s_i + \int_{w_i}^{\beta} \Phi(t)\, dt + 1 - \beta} \leq t_i$$

$$[w_{i+1}] \qquad \Phi(w_{i+1}) = q_{i+1}$$

$$[\mathrm{u}] \qquad w_i \leq w_{i+1} \leq 1$$

for each rate interval $[q_i, q_{i+1})$.

As explained in Section 4, our algorithm builds a function Φ and values w_i in an iterative way. That is, it processes each set of constraints iteratively, and at each step $j \in [1, l]$ it builds a function Φ_j and computes values $w_1, \ldots, w_{j+1}$ which satisfy the sets of constraints for all intervals $[q_i, q_{i+1})$ with $i \leq j$. Each function Φ_j and the new values $w_1, \ldots, w_{j+1}$ are a function of Φ_{j-1} and the previous values $w_1, \ldots, w_{j+1}$.

We explain an iteration of this process. Suppose that the algorithm is at a step where it has computed Φ_{j-1} and values $w_1, \ldots, w_j$ as to satisfy the sets of constraints for the intervals $[q_i, q_{i+1})$ with $i < j$. Constraint $[\beta]$ requires us to guarantee a ratio of at least t_j for every sequence whose maximum rate is in $[q_j, q_{j+1})$. We derive a function which achieves a ratio *equal* to t_j for such sequences. The equality is sought, instead of the inequality, in order to minimize utilization. Intuitively, enforcing a ratio smaller than t_j would force the algorithm to exchange more money to achieve a bigger profit. Thus the following constraint

$$\forall \beta \in [w_j, w_{j+1}) : \frac{\Phi(\beta)}{s_j + \int_{w_j}^{\beta} \Phi(t)\, dt + 1 - \beta} = t_j,$$

from which we can obtain the differential equation:

$$\dot{\Phi} = t_j \cdot \Phi - t_j, \tag{B.1}$$

which is a separable first order differential equation. We can hence find the unique solution

$$\Phi(\beta) = C \cdot e^{t_j \cdot \beta} + 1.$$

We then apply constraint [β], for an arbitrary $\beta \in [w_j, w_{j+1})$, so to find the value of the constant C, which yields

$$\boxed{\Phi(\beta) = (t_j \cdot (s_j + 1 - w_j) - 1) \cdot e^{t_j \cdot (\beta - w_j)} + 1} \tag{B.2}$$

The obtained function is the unique solution to such an equation. We denote $\rho_j = t_j \cdot (s_j + 1 - w_j)$.

We then use constraint [w_{j+1}] to find an expression for w_{j+1}:

$$\boxed{w_{j+1} = \frac{1}{t_j} \ln \left(\frac{q_{j+1} - 1}{\rho_j - 1} \right) + w_j} \tag{B.3}$$

Note that $\Phi(w_j) = \rho_j$. There are two cases to be analyzed.

First, if $\rho_j > q_j$, then we can define Φ_j as follows:

$$\Phi_j(w) = \begin{cases} \Phi_{j-1}(w) & \text{if } w \in [1, w_j) \\ (t_j \cdot (s_j + 1 - w_j) - 1) \cdot e^{t_j \cdot (\beta - w_j)} + 1 & \text{if } w \in [w_j, w_{j+1}), \end{cases}$$

where w_{j+1} is defined in (B.3). We say that we extend the previous Φ_{j-1}. This scenario materializes when the algorithm has achieved a profit s_j, which allows it to not exchange while observing rates in $[q_j, \rho_j]$ and still remain t_j-competitive. This occurs when $t_j > t_{j-1}$, hence it occurs for the increasing part of the profile.

On the other hand, $\rho_j < q_j$, if $t_j < t_{j-1}$. If this case occurs, the algorithm has not obtained a sufficient profit to be t_j-competitive when presented with the sequence which continuously increases from 1 to q_j, which is the worst-case sequence as stated in Remark 2.1. As we will show in the proof of Theorem 4.1 w_j is the least utilization that can be spent so to satisfy every set of constraints $[q_k, q_{k+1})$ with $k < j$. To enforce a ratio of t_j and still minimize utilization, the algorithm must exchange a bigger amount when rate q_j is revealed, since exchanging more at a lower rate would lead to a larger utilization. To guarantee a ratio of t_j for the continuous increasing sequence, the algorithm should trade an amount equal to $w'_j - w_j$, where w'_j is obtained from:

$$\frac{q_j}{s_j + q_j \cdot (w'_j - w_j) + 1 - w'_j} = t_j$$

and leads to

$$w'_j = \frac{q_j - t_j \cdot (s_j - w_j q_j + 1)}{t_j \cdot (q_j - 1)}.$$

We now wish to extend function Φ_{j-1}, obtained in the previous iteration, so as to satisfy all constraints for interval $[q_j, q_{j+1})$. Let $s'_j = s_j + q_j \cdot (w'_j - w_j)$, which is the profit obtained by the OTA in the worst case where the maximum rate is q_j. We may express this problem by a new set of constraints, which are:

$$
\begin{aligned}
&[\beta] && \forall \beta \in [w'_j, w_{j+1}) : \frac{\Phi(\beta)}{s'_j + \int_{w'_j}^{\beta} \Phi(t)\, dt + 1 - \beta} \le t_j, \\
&[w_{j+1}] && \Phi(w_{j+1}) = q_{j+1}, \\
&[\text{u}] && w'_j \le w_{j+1} \le 1.
\end{aligned}
$$

Note that this set of constraints is the same as the ones we started with, but s_j was replaced by s'_j and w_j by w'_j. Hence, the Φ and w_{j+1} which satisfy the constraints and minimize w_{j+1} are:

$$\Phi(\beta) = (t_j \cdot (s'_j + 1 - w'_j) - 1) \cdot e^{t_j \cdot (\beta - w'_j)} + 1, \tag{B.4}$$

$$w_{j+1} = \frac{1}{t_j} \ln \left(\frac{q_{j+1} - 1}{t_i \cdot (s' + 1 - w'_i) - 1} \right) + w'_j. \tag{B.5}$$

We can now proceed with the proof for Theorem 4.1.

Proof of Theorem 4.1. As stated in Remark 2.1, every online strategy will exchange on rates which are best-seen so far. We can hence state every strategy as an OTA. It suffices then to prove the following: There exists an OTA which respects F if and only if PROFILE terminates with a value $w_{l+1} \leq 1$.

Let F be a performance profile. The if direction follows directly from the design of PROFILE. It suffices to observe that the obtained function Φ_l can be used as the threshold function for an OTA which respects the profile F.

To prove the only if direction, we will prove that every w_i obtained by PROFILE is the least utilization needed to satisfy all sets of constraints for intervals $[q_k, q_{k+1})$ for $k < i$. In other words, we will prove that if A is an OTA, which respects F, defined by Φ, and where $w'_1, \ldots, w'_{l+1}$ are the respective utilization levels reached by A when observing rates $q_1, \ldots, q_{l+1}$, i.e: $\Phi(w'_i) = q_i$ for each $i \in [1, \ldots, l+1]$, then $w_i \leq w'_i$. This statement follows, once again, from the design of PROFILE. By replacing the inequality constraint in [β] by an equality, we manage to achieve a ratio which is exactly the one demanded by the profile, hence reserving budget for futures rates. PROFILE obtains a function Φ_l which enforces, for each $i \in [1, l]$ and for each $q \in [q_i, q_{i+1})$ the equation:

$$\frac{q}{\int_1^{\Phi_l^{-1}(q)} \Phi_l(u)du + 1 - \Phi_l^{-1}(q)} = t_i.$$

We conclude that PROFILE minimizes utilization while satisfying every set of constraints, thus proving the theorem. □

Figure 3 illustrates PROFILE. Here we observe that for the increasing part of the profile, Φ_i with $i \in [4, 7]$ extends Φ_{i-1} with an exponential function starting at w_i, where $\Phi_i(w_i) > \Phi_{i-1}(w_i)$. Here the vertical "jumps" reflect the less stringent requirement in the increasing part (we can afford to reserve our budget for later). For the decreasing part of the profile, Φ_i with $i \in [1, 3]$ extends Φ_{i-1} with an exponential function starting at $w'_i > w_i$ (line 9 in the statement) where $\Phi_i(w'_i) = \Phi_{i-1}(w_i)$, which is reflected in the presence of straight lines in Figure 3.

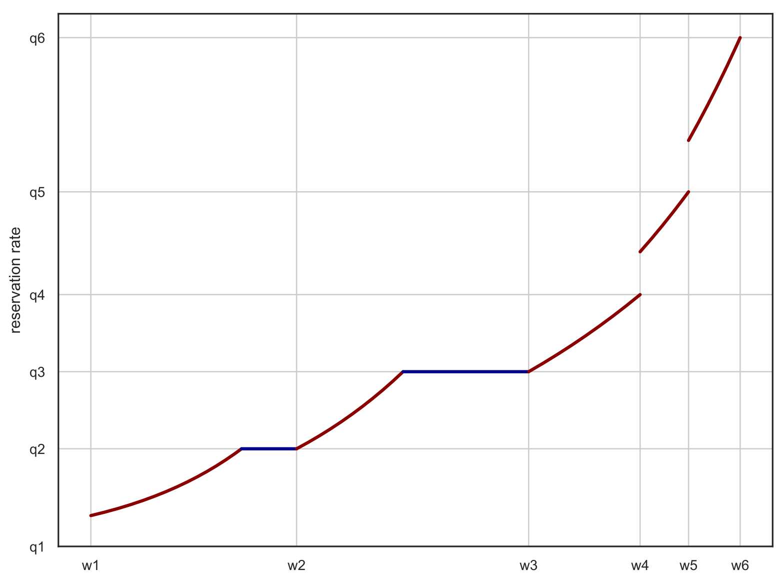

Figure 3: An illustration of PROFILE. Here the profile F is as follows: $F([1, 20) = 7, F([20, 35]) = 5$, $F([35, 50]) = 3$, $F([50, 70]) = 3.5$, and $F([70, 100]) = 4$

.

C Details from Section 5

In this section, we detail the calculations that lead to the value w_{i+1}, which is the maximum an online algorithm can spend on rate p_i while ensuring r-robustness.

The aforementioned w_{i+1} is the solution to the following optimization problem:

$$\begin{aligned}
&\max && w && (O_i)\\
&\text{subj. to} && && \\
&[\beta] && \forall \beta \in [w,1) : \frac{\Phi(\beta)}{s_i + p_i \cdot (w - w_i) + \int_w^\beta \Phi(t)\,dt + 1 - \beta} = r, && \\
&[M] && \Phi(1) \geq M, && \\
&[\text{u}] && w_i \leq w \leq 1. &&
\end{aligned}$$

From constraint $[\beta]$, we do the same analysis as in B to find $\Phi(\beta) = C \cdot e^{r\beta} + 1$. Once again, to find the constant C we use constraint $[\beta]$ for an arbitrary value $\beta \in [w_{i+1}, 1]$, which leads to:

$$\Phi(\beta) = \big(r \cdot (s_i + 1 - p_i w_i + w_{i+1} \cdot (p_i - 1)) - 1\big) \cdot e^{r\cdot(\beta - w_{i+1})} + 1.$$

We then use constraint [M] to obtain an upper bound on w_{i+1}:

$$\big(r \cdot (s_i + 1 - p_i w_i + w_{i+1} \cdot (p_i - 1)) - 1\big) \cdot e^{r\cdot(1 - w_{i+1})} + 1 \geq M,$$

which leads to:

$$w_{i+1} \leq 1 - \frac{1}{r} \ln\left(\frac{M-1}{r(s_i + 1 - p_i w_i + w_{i+1}(p_i - 1) - 1)}\right).$$

Thus the largest value of w_{i+1} is the root of the equation

$$w_{i+1} = 1 - \frac{1}{r} \ln\left(\frac{M-1}{r(s_i + 1 - p_i w_i + w_{i+1}(p_i - 1) - 1)}\right),$$

which can be solved using numerical methods. Let ρ be the reservation rate for utilization w_{i+1}, then

$$\rho = \Phi(w_{i+1}) = r \cdot (s_i + 1 - p_i w_i + w_{i+1} \cdot (p_i - 1)).$$

If $\rho > M$, then the algorithm has achieved a sufficient profit to guarantee r-robustness independently of future rates. Hence, to maximize w_{i+1}, we can safely set it to 1. However, if $\rho < M$, then constraint $[M]$ was saturated, and the algorithm will achieve a performance ratio of r for every sequence which grows continuously from ρ until a rate $p^* \in [\rho, M]$. Moreover, for every sequence whose maximum rate $p^* \in [p_i, \rho)$ the algorithm will have a performance ratio smaller than r.

As explained in Appendix B using constraint $[\beta]$ with an equality allows us to guarantee a performance ratio of r minimizing utilization. Observe that to maximize w_{i+1} we need to minimize the left-over budget to remain r-robust in the future. We can hence conclude that $w_{i+1} - w_i$ is indeed the largest amount of money we can exchange at rate p_i and remain r-robust.

We will next provide the proof for Theorem 5.1.

Proof of Theorem 5.1. We are to prove that ADA-PO is Pareto-Optimal and dominates every other Pareto-Optimal algorithm on any sequence σ.

First, we will prove that ADA-PO is Pareto-Optimal. Let r be a a robustness requirement, and $c(r)$ the respective consistency. To start with, we prove that ADA-PO is r-robust. Consider first the (easy) case where $p^* < \hat{p}$ then ADA-PO assures a performance ratio of r using the threat-based approach.

Consider then the (harder) case in which $p^* > \hat{p}$. Let p_i be the first rate above $\hat{p}$ and w_{i+1},Φ_i be the respective solution to problem O_i. We must prove that no matter how the sequence continues ADA-PO achieves a performance ratio of at least r. If $\Phi(w_{i+1}) \geq M$ then a performance ratio of r is guaranteed, due to $\frac{M}{s_{i+1}+1-w_{i+1}} \leq r$, from constraint $[\beta]$. Suppose then $\Phi_i(w_{i+1}) < M$, then by constraints [M] and [u] we know that $w_{i+1} < 1$. When the next rate $p_{i+1} > p_i$ is revealed the same analysis can be applied. We thus obtain a non-decreasing sequence of reservation rates $\Phi_j(p_j)$

for $j > i$. For each rate, problem O_i is solved. Note that the feasibility of problem O_i with rate p_i implies the feasibility of the problem O_i with the next rate as shown by the next analysis. Namely, if $p_i \leq \Phi(p_{i-1})$ then $w = w_i$, $\Phi_i = \Phi_{i-1}$ is a solution, and if $p_i > \Phi(p_{i-1})$, then $w = \Phi_{i-1}^{-1}(p_i)$, $\Phi_i = \Phi_{i-1}$ is as well. Furthermore, both cases lead to a performance ratio of at least r in case the next rate equals 1 and is the last rate. We hence conclude, that either one of the reservation rates is greater or equal than M or ADA-PO successfully achieves a performance ratio of r for each rate ($w_i < 1$ was a solution for each problem). We conclude then that ADA-PO is r-robust.

We will now prove that ADA-PO is $c(r)$-consistent. We must prove that for every error-free sequence the performance ratio is at most $c(r)$. Let A' be any Pareto-Optimal algorithm. When observing rates below $\hat{p}$, ADA-PO follows the threat-based policy, hence for every error-free sequence, its budget is at least the same as A' when a rate equal to $\hat{p}$ is exhibited. Then by solving the optimization problem, ADA-PO exchanges the most it can in order to remain r-robust, a larger amount would make the problem infeasible. In other words, there would not exist a function Φ satisfying the constraints, and the continuously increasing function from $\hat{p}$ to M will lead to a performance ratio bigger than r. Hence, no other algorithm could achieve a better profit. We conclude that ADA-PO is $c(r)$-consistent.

We finally prove that ADA-PO dominates A'. By the previous analysis, when observing the first rate above the prediction, ADA-PO has a budget at least the budget than A'. As ADA-PO exchanges the most it can to remain r-robust, it will obtain a next utilization which is equal or smaller than A', hence achieving a better profit, because A' exchanged the same or less at lower rates. If A' has behaved the same as ADA-PO, then this process repeats for every following rate. We conclude then that ADA-PO dominates or performs equally to A'. □

Remark C.1. To conclude we offer an intuitive explanation of dominance. If the maximum rate of the sequence is below the prediction, then ADA-PO's profit will be smaller or equal than any other Pareto-Optimal algorithm. Its profit will be equal if the sequence is a continuously increasing one. Moreover, for the first rate equal or greater than the prediction, its profit will be greater or equal than any other Pareto-Optimal algorithm. By definition of dominance, while observing rates above the prediction, either the two profits will be equal, or ADA-PO's profit is larger, unless the Pareto-Optimal algorithm attained a smaller profit at an earlier rate.

D Profile-based contract scheduling

In this section, we discuss another application of our profile-based framework of Section 3. Specifically, we focus on another well-known optimization problem that has been studied under learning-augmented settings, namely contract scheduling. In its standard variant, the problem consists of finding an increasing sequence $X = (x_i)_{i=0}^{\infty}$ which minimizes the *acceleration ratio*, formally defined as

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

where $\ell(X,T)$ denotes the *largest* contract completed by T in X, namely

$$\ell(X,T) = \max_j \{x_j : \sum_{i=0}^{j} x_i \leq T\}.$$

Contract scheduling is a classic problem that has been studied under several settings. In its simplest variant stated above, the optimal acceleration ratio is equal to 4 [37], but many more complex settings have been studied in the literature; see [7] and references therein. In this section we are interested in the learning augmented setting introduced in [7] in which there is a *prediction* τ on the interruption time T. The prediction *error* is defined as $\eta = |T - \tau|$. In this context, the consistency $c(X)$ of schedule X is defined as

$$c(X) = \frac{\tau}{\ell(X,\tau)},$$

whereas its robustness is defined as

$$r(X) = \sup_{T \geq 1} \frac{T}{\ell(X,T)},$$

i.e., the worst-case performance of X, assuming adversarial interruptions. Since the latter occur arbitrarily close to the completion time of any contract, we obtain an equivalent interpretation of the robustness as

$$r(X) = \sup_{i \geq 1} \frac{\sum_{j=0}^{i} x_j}{x_{i-1}}.$$

In [7] it was shown that the optimal consistency of a 4-robust schedule is equal to 2. However, as proven in [5], any such schedule suffers from brittleness. Namely, for any $\epsilon > 0$, there exists a prediction τ and an actual interruption time T such that $|T - \tau| = \epsilon$, and any 4-robust and 2-consistent schedule satisfies $\ell(X, T) \leq \frac{T+\epsilon}{4}$.

In the remainder of this section we will show how to use our framework of profile-based performance so as to remedy this drawback. For definiteness, and to illustrate the application of the techniques, we consider the requirement that the performance of the schedule degrades linearly as a function of the prediction error. Namely, suppose that we require that $f(X, T) := T/\ell(X, T)$ be respect a profile F_ϕ, where the latter is defined as a symmetric, bilinear function that is decreasing for $T \leq \tau$, and increasing for $T \geq \tau$, with slope ϕ, as illustrated in Figure 4. This profile is chosen by the schedule designer, and the angle ϕ captures the "smoothness" at which the schedule is required to degrade as a function of the prediction error.

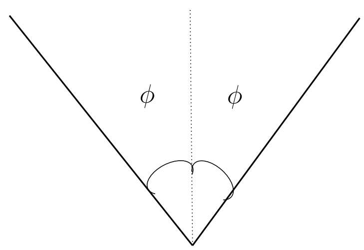

Figure 4: An illustration of the profile F_ϕ.

More specifically, for a given prediction τ, and a profile F_ϕ as above, we are interested in finding the best extension of F_ϕ such that there exists a 4-robust schedule that respects the extension. We can thus define the analytical concept of *consistency according to* F_ϕ as

$$c_{F_\phi} := \sup_\tau \inf_T \frac{T}{\ell(X, T)} : \ X \text{ respects } F_\phi.$$

The following theorem states our main result.

Theorem D.1. Given a profile F_ϕ and a prediction τ on an interruption time, we can compute a 4-robust schedule that respects F_ϕ and has optimal consistency according to F_ϕ.

Proof. We will assume that X if of the form $(\lambda 2^i)_{i \in \mathbb{Z}}$. This is not a limiting assumption, as discussed in [5], and its purpose is to simplify the calculations. Since any 4-robust schedule is of the above form [5], it will suffice to compute a λ that satisfies the constraints of our problem, and the result will follow.

Recall that $f(X, T)$ denotes the function $T/\ell(X, T)$. By definition, for every $i \in \mathbb{N}$, $f(X, T)$ is a linear, increasing function of T function in the interval $I_k = [T_k, T_{k+1}] = [\lambda 2^k, \lambda 2^{k+1}]$, with smallest value equal to 2, and largest value equal to 4.

With the above observation in mind, for a given, fixed λ, let k be such that $\tau \in I_{k+1}$, i.e., we have that $\ell(X, \tau) = \lambda 2^k$. Define $\alpha \in [1, 2]$ to be such that $\tau = \alpha T_k$, and note that by construction, α is a function of λ. Moreover

$$f(X, \tau) = \frac{\tau}{\lambda 2^k} = \frac{\alpha T_k}{\lambda 2^k} = \frac{\alpha \lambda 2^{k+1}}{\lambda 2^k} = 2\alpha, \tag{D.2}$$

which implies that it suffices to compute α, then λ must be chosen so that $\lambda = 2^{\{\log(2\alpha)\}}$, where $\{x\}$ denotes the fractional part of x.

In order to minimize f, subject to X respecting the profile, λ must be chosen such that one of the two cases occur, which we analyze separately.

Case 1. The profile F_ϕ has a unique intersection point with f at $T = \tau$, and moreover $F(T_k + \epsilon) = 4$, for infinitesimally small $\epsilon > 0$. This situation is illustrated in Figure 5. For this case to arise, and for the schedule to be consistent with F, it must be that

$$\tan(\frac{\pi}{2} - \phi) \geq \frac{4-2}{T_{k+1} - T_k} = \frac{2}{T_k} = \frac{2\alpha}{\tau}. \tag{D.3}$$

It must then be that $f(X, \tau) + \frac{\tau - T_k}{\tan \phi} = 4$, hence

$$4 - \rho(1 - \frac{1}{\alpha}) = 2\alpha, \text{ where } \rho = \frac{\tau}{\tan \phi}.$$

Solving the above equality for α minimizes f, by means of (D.2). We obtain that

$$\alpha = \frac{1}{4}(\sqrt{\rho^2 + 16} - \rho + 4) \text{ and } f(X, \tau) = 2\alpha,$$

subject to the condition (D.3).

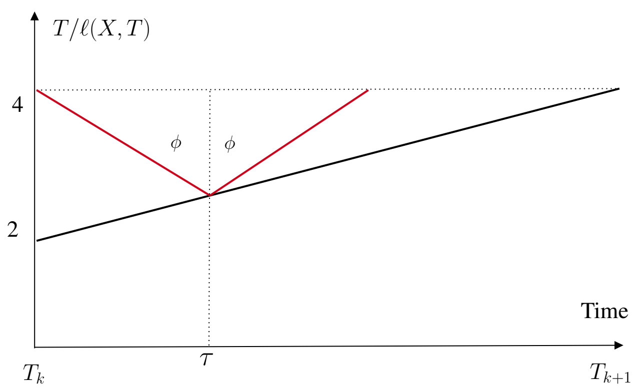

Figure 5: An illustration of Case 1.

Case 2. This case occurs if the condition in Case 1 does not apply. The profile F_ϕ is such that $F(T_k + \epsilon) = F(T_{k+1} - \epsilon)4r$, for infinitesimally small $\epsilon > 0$. This situation is illustrated in Figure 6. For this case to arise, and for the schedule to respect F_ϕ it must be that $\tau = \frac{T_{k+1}+T_k}{2} = \frac{3}{2}\frac{\tau}{\alpha}$, hence $\alpha = 3/2$. In this case, we obtain that

$$f(X, \tau) = 4 - \frac{T_{k+1} - \tau}{\tan \phi} = 4 - \rho, \text{ where } \rho = \frac{\tau}{\tan \phi}.$$

$\square$

We observe that in both cases in the analysis of Theorem D.1 we obtain that $f \in (2, 4]$, as a function of τ and ϕ. This result makes intuitively sense, since X is 4-robust, and the smallest consistency is equal to 2 (when $\phi \to 0$).

E Further experimental analysis

To further quantify the performance difference between the two algorithms, PROFILE and PO, we performed additional experiments. Specifically, we used a list of the last 20,000 minute-exchange rates of BTC to USD, so as to create 20 different sequences, each with its own prediction, using the same method as in Fig 2c. For each sequence, we computed the average improvement over PO for rates in the interval of interest $[0.9\hat{p}, 1.1\hat{p}]$. Figure 7 depicts this average for each of the 20 sequences. We observe that for the sequences in which PROFILE outperforms PO (12 out of 20), the improvement ranges from roughly 15% to 30%, whereas PO outperforms PROFILE in 8 out of 20 sequences, by a factor that is at most 10%, roughly.

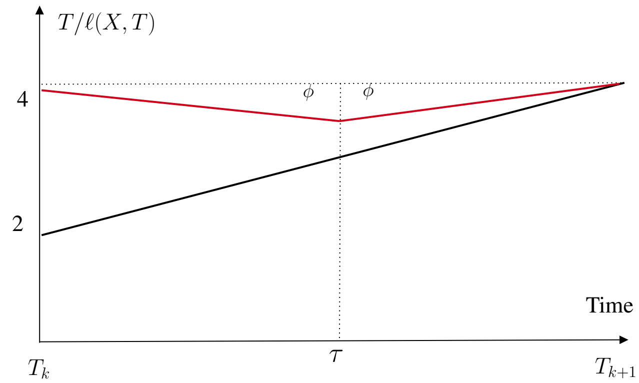

Figure 6: An illustration of Case 2.

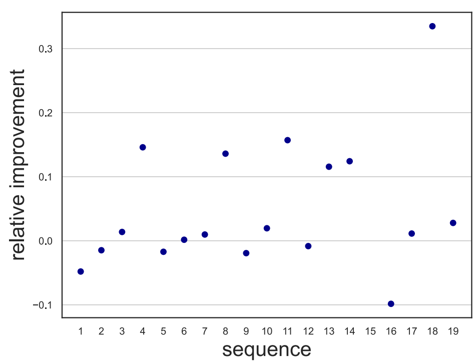

Figure 7: Average ratio improvement of PROFILE over PO

1043 # F Computational setup

1044 The experiments are reproducible on any computer with the experimental setup described in the README file. They do not require any memory or computational powe beyond the standard requirements. They run typically on few milliseconds.