

Optimization Methods and Software

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/goms20

Optimal inexactness schedules for tunable oracle-based methods

Guillaume Van Dessel & François Glineur

To cite this article: Guillaume Van Dessel & François Glineur (2024) Optimal inexactness schedules for tunable oracle-based methods, Optimization Methods and Software, 39:3, 664-698, DOI: 10.1080/10556788.2023.2296982

To link to this article: https://doi.org/10.1080/10556788.2023.2296982

Published online: 05 Feb 2024.

Submit your article to this journal

Article views: 15

View related articles

View Crossmark data

Full Terms & Conditions of access and use can be found at
https://www.tandfonline.com/action/journalInformation?journalCode=goms20

OPTIMIZATION METHODS & SOFTWARE
2024, VOL. 39, NO. 3, 664–698
https://doi.org/10.1080/10556788.2023.2296982

Taylor & Francis
Taylor & Francis Group

Check for updates

Optimal inexactness schedules for tunable oracle-based methods

Guillaume Van Dessel[a] and François Glineur[a,b]

[a]UCLouvain, ICTEAM (INMA), Louvain-la-Neuve, Belgium; [b]UCLouvain, CORE, Louvain-la-Neuve, Belgium

ABSTRACT

Several recent works address the impact of inexact oracles in the convergence analysis of modern first-order optimization techniques, e.g. Bregman Proximal Gradient and Prox-Linear methods as well as their accelerated variants, extending their field of applicability. In this paper, we consider situations where the oracle's inexactness can be chosen upon demand, more precision coming at a computational price counterpart. Our main motivations arise from oracles requiring the solving of auxiliary subproblems or the inexact computation of involved quantities, e.g. a mini-batch stochastic gradient as a full-gradient estimate. We propose optimal inexactness schedules according to presumed oracle cost models and patterns of worst-case guarantees, covering among others convergence results of the aforementioned methods under the presence of inexactness. Specifically, we detail how to choose the level of inexactness at each iteration to obtain the best trade-off between convergence and computational investments. Furthermore, we highlight the benefits one can expect by tuning those oracles' accuracy instead of keeping it constant throughout. Finally, we provide extensive numerical experiments that support the practical interest of our approach, both in *offline* and *online* settings, applied to the Fast Gradient algorithm.

ARTICLE HISTORY
Received 30 December 2022
Accepted 15 December 2023

KEYWORDS
Inexact oracles; tunable accuracy; optimal schedules; first-order algorithm

1. Introduction

Typical iterative optimization schemes rely on key ingredients often referred to as *oracles* [6]. In what concerns continuous optimization, with respect to both convex and nonconvex realms, the vast majority of the papers use two distinctive types of oracles at each iteration, namely the *informative* and *computational* ones [22].

(I) **Informative**: One assumes the possibility to obtain differential information about the objective (zero, first, . . . , higher-order) at successive query points.
(II) **Computational**: One assumes the ability to update sequences of iterates following some rules, often involving the resolution of easy or, at least, not too complicated, subproblems.

CONTACT Guillaume Van Dessel guillaume.vandessel@uclouvain.be UCLouvain, ICTEAM (INMA), 4 Avenue Georges Lemaître, Louvain-la-Neuve, Belgium

© 2024 Informa UK Limited, trading as Taylor & Francis Group

By default, one considers implicitly that both oracles yield *error-free* outputs. Nevertheless there exist cases for which such commodity appears unreasonable, e.g. when the objective value stands as the result of a non-trivial optimization problem (I) [27] or when one cannot solve exactly subproblems in (II) [14]. Unfortunately, more and more problems of practical interest exhibit a structure that does not allow for *exact* oracles. Therefore there has been put a lot of efforts over the past years in dealing with inexactness about *informative* and *computational* oracles in order for widely spread algorithms (Bregman Proximal Gradient, Prox-Linear, their accelerated variants, etc.) [5,21,26] to remain applicable in such scenarios. As argued in the previous paragraph, a common source of inexactness is due from the necessity of numerically solving auxiliary non-trivial optimization problems to produce oracles' outputs.

Example 1.1 (saddle-point problems): For instance, [3] analysed the Gradient Method (GM) that allowed for what they defined as (δ, L, μ) inexact oracles. Consider saddle-point problems of the type

$$F^* = \min_{x\in\mathbb{R}^d} \left\{ F(x) := \max_{u\in\mathbb{R}^n} G(u) + \langle Au, x\rangle \right\} > -\infty, \tag{1}$$

where G is $L(G)$ smooth and $\mu(G) > 0$ strongly-concave and $A \in \mathbb{R}^{d\times n}$ is a matrix. They showed that F is $L(F) = \frac{\lambda_{\max}(AA^T)}{\mu(G)}$ smooth, $\mu(F) = \frac{\lambda_{\min}(AA^T)}{L(G)}$ strongly-convex and its gradient is given as $\nabla F(x) = Au^*(x)$ where $u^*(x)$ is the exact minimizer of

$$\max_{u\in\mathbb{R}^n} G(u) + \langle Au, x\rangle. \tag{2}$$

In the general case, (2) cannot be solved exactly. Nevertheless, it can be approximately solved up to $\delta > 0$ global accuracy quite efficiently, e.g. by dedicated accelerated first-order methods [13]. That is, instead of $u^*(x)$, one can provide $u_x \in \mathbb{R}^n$ such that

$$G(u^*(x)) + \langle Au^*(x), x\rangle - (G(u_x) + \langle Au_x, x\rangle) \le \delta \tag{3}$$

and use approximate information in GM: $\tilde{F}(x) = G(u_x) + \langle Au_x, x\rangle - \delta$, $\nabla\tilde{F}(x) = Au_x$. Thereby and luckily enough, it happens that one can *tune* the quality of oracles by adequately choosing the amount of computational time spent on these problems. Looking back at the above example, it is well known that the number ω of iterations one must undertake to solve (2) up to δ accuracy (3) scales as $\kappa(G)^{-\frac{1}{2}}\ \log(\delta^{-1})$ where $\kappa(G) = \mu(G)/L(G)$. After N steps of GM involving a sequence of such inexact gradients with parameters $\{\delta_k\}_{k=0}^{N-1}$ at iteration $k = 0, \ldots, N-1$ while using a constant stepsize $L^{-1} = (2L(F))^{-1}$, it is proven in [3] that for

$$\hat{x}_N = \frac{\sum_{k=0}^{N-1} (1-\kappa)^{N-1-k} x_k}{\sum_{k=0}^{N-1} (1-\kappa)^{N-1-k}}$$

the following guarantees held:

$$F(\hat{x}_N) - F^* \le \frac{M(N) + 3\sum_{k=0}^{N-1} (1-\kappa)^{N-1-k}\delta_k}{\sum_{k=0}^{N-1} (1-\kappa)^{N-1-k}} \tag{4}$$

with $\kappa = 4^{-1}\kappa(G), M(N) = 2^{-1} LR^2 (1-\kappa)^N$ and R initial distance to a minimizer of (1). One observes the additive impact of such inexactness on GM's convergence.

It is worth noting that inexactness also naturally occurs in the context of stochastic gradient methods. Within such framework, one can *tune* stochastic gradient and/or Hessian's bias by averaging more or less sample gradients and/or Hessians [7,9].

Despite the fact that convergence guarantees under a different level of inexactness at each iteration are well-established in the literature, very few works devise specific inexactness schedules as in [4,25]. This can provably constitute a miss of opportunity. Akin to our introductory example, let us assume that we dispose of relations linking the quality of oracles (δ) with the computational efforts invested in the creation of their output (ω). It is possible to retrieve optimal inexactness schedules taking into account the *trade-off* between the computational price and the harm in terms of convergence guarantees, e.g. (4), of the prescribed oracle precision. We detail our optimality criteria in Section 3. Informally, we aim in this work at answering the question: *How can we make the most of a computational budget when using optimization methods dealing with oracle inexactness?*

1.1. Related work

At first sight and as is often the case in mathematical optimization research, the goal of our work can be summed up simply as an improvement of a worst-case convergence bound. Indeed, under the models we motivate in Section 2 and the assumptions we introduce in Section 3, we solve a specific instance of *non-linear allocation* problems to provide our enhanced inexactness schedules in terms of convergence guarantees per computational cost unit.

How straightforward might it sound, to the best of our knowledge there actually exists not so many previous works on that subject, i.e. *trade-off* optimality for iterative algorithms dealing with controllable inexactness. We contrast, however, this last sentence by reassuring that a bunch of new papers propose criteria involving parameter fixed *relative* inexactness (see [10,18] and references therein). In opposition with the framework of *absolute* inexactness, computational costs to achieve *relative* inexactness are by nature prone to uncertainty and worst-case guarantee dependencies on this parameter often appear quite opaque. One usually leaves the relative inexactness parameter as fixed to a conservatively low wished terminal accuracy. Authors of [13] studied a mix between *relative* and *absolute* inexactness, highlighting the positive impact of incorporating *absolute* oracle accuracy.

With a similar approach to ours, [12] focus on (Accelerated) Proximal Gradient algorithms as analysed by [20] in their seminal paper. Unlike us, they do not show explicit closed-forms, i.e. analytical value for δ_k, method's level of inexactness at iteration $k = 0, \ldots, N-1$ as in (4), and their results remain mainly theoretical.

More recently, [25] comes up with specific schedules for the inexactness in the auxiliary subproblems they deal with to lower the overall computational load after $N \in \mathbb{N}$ iterations. Our more general results encompass theirs if we were concerned with the same context of inexact Augmented Lagrangian. As a byproduct of an asymptotical analysis, i.e. $N \to \infty$, [24] whose follow-up resides in [13], suggest to use inexactness schedules that decrease sufficiently fast to maintain (up to a logarithmic factor) the rate of convergence of the *error-free*, i.e. *exact*, counterpart of the algorithms at scope. In [4] were tried *online* (adaptive) as well as purely *offline* schedules in what concerns the accuracy of the inexact higher-order tensor steps. Both last works provide *non-constant* inexactness schedules but none of them brings into balance (inexact) oracles' computational complexities.

1.2. Contributions

Let us consider an iterative algorithm involving controllable inexactness, e.g. GM as depicted in the previous paragraph. We can state our main contributions as follows.

- First, considering fixed the number $N \in \mathbb{N}$ of inexact oracle calls, we propose a systematic *offline* procedure to devise the amount δ_k of inexactness to adopt at each iteration $k \in \{0, \dots, N-1\}$ based on an oracle cost model and algorithm's guarantees. To that end, we solve a rather general *non-linear assignment* problem, result of independent interest. Building upon this, we present closed-form results about the optimal inexactness schedules for a broad class of oracle cost models, directly inspired from practical scenarios.
- Second, we propose an *online* heuristic extension in which neither N nor the overall computational budget allocated must be fixed beforehand.
- Third, we conduct numerical experiments that sustain the validity of our approach, either *offline* or *online*, by comparing it against (a) *constant* inexactness schedules and (b) *non-constant* inexactness schedules from the literature [13,24]. We emphasize on the fact that our strategy is fully implementable.

1.3. Outline

At the end of the present section, we clarify our notations and define the useful concept of *descending rank*. In Section 2 are motivated and then explained the concept of Tunable (Inexact) Oracles, i.e. we develop the models of costful (inexact) oracles and convergence under an inexactness schedule which we illustrate thanks to three examples serving as guidelines. We then proceed to Section 3 in which we show all the contributions teased just above. This section represents the main content of this paper. We conclude by quantifying the computational savings of our approach on the aforementioned guideline examples in Section 4.

1.4. Preliminaries

We introduce some handy notations and a definition extensively used in this paper.

Sets $\mathbb{N} = \{1, \dots, \infty\}$ will refer to the set of strictly positive integers. $\mathbb{R}_+$ (respectively $\mathbb{R}_-$) contains all the non-negative (respectively non-positive) real numbers. We denote $\mathbb{R}_{++} = \mathbb{R}_+ \backslash \{0\}$ and $\mathbb{R}_{--} = \mathbb{R}_- \backslash \{0\}$. Let $n \in \mathbb{N}$, we define $[n] = \{0, \dots, n-1\}$.

Sequences Our enumerating indices start at 0. We write column vectors $v \in \mathbb{R}^n$ as $v = (v_0, \dots, v_{n-1})^T$. Depending on the context, we will equivalently write $v \equiv \{v_k\}_{k=0}^{n-1}$.

Shorthands $\mathbf{e} \in \mathbb{R}^n$ stands as the vector of size n full of 1. For any $\mathcal{S} \subseteq [n]$, $\mathbf{e}_\mathcal{S}$ is defined as follows: $(\mathbf{e}_\mathcal{S})_k = 1$ if $k \in \mathcal{S}$ and 0 otherwise. On the other hand, $\mathbf{0} \in \mathbb{R}^n$ stands as the vector of size n full of 0, i.e. $\mathbf{0} = \mathbf{e}_\emptyset$. For any $v \in \mathbb{R}^n$, we define the subvector $(v)_\mathcal{S}$ of size $|\mathcal{S}| \leq n$ whose values are taken from v at indices in $\mathcal{S}$.

Operations on vectors Let $p \in \mathbb{R}$ and $u, v \in \mathbb{R}^n$. We proceed to entry-wise operations like $v^p = (v_0^p, \dots, v_{n-1}^p)^T$, $u \odot v = (u_0 \cdot v_0, \dots, u_{n-1} \cdot v_{n-1})^T$, $u^T v = \sum_{k=0}^{n-1} u_k \cdot v_k$ and $|v| =$

$(|v_0|, \ldots, |v_{n-1}|)^T$ More generally, applying any operator $o : \mathbb{R} \to \mathbb{R}$ in an element-wise fashion on a vector v is authorized: $o(v) = (o(v_0), \ldots, o(v_{n-1}))^T$.

p-norms For any $p \geq 1$ and any $v \in \mathbb{R}^n$, the p-norm of v reads $||v||_p = (\mathbf{e}^T |v|^p)^{\frac{1}{p}}$.

As usual, we extend the notation with $p = \infty$ and set $||v||_\infty \triangleq \max\{|v_k| \,|\, k \in [n]\}$.

In what concerns matrices, the p-norm of any element $V \in \mathbb{R}^{n \times d}$ translates to

$$||V||_p = \sup_{u \neq \mathbf{0}} \frac{||Vu||_p}{||u||_p}.$$

Unless stated otherwise, we understand norms and distances as Euclidean ones ($p = 2$) throughout this paper, i.e. $||v|| := ||v||_2$ and $||V|| := ||V||_2$.

Lipschitz continuity $g : \mathbb{R}^n \to \mathbb{R}^d$ is L Lipschitz (continuous) with respect to a p-norm if for any $y, x \in \text{dom}(g)$, $||g(y) - g(x)||_{p^*} \leq L\,||y - x||_p$ with $p^* = p \cdot (p-1)^{-1}$.

Inequalities Let $\mathcal{L} : \Xi \subseteq \mathbb{R} \to \mathbb{R} \cup \{\infty\}$, $\mathcal{J} : \Xi \subseteq \mathbb{R} \to \mathbb{R} \cup \{\infty\}$ be two functions. Let $\text{dom}(\mathcal{L}) = \text{dom}(\mathcal{J})$, we consider the following:

$$\mathcal{L} \succeq \mathcal{J} \Leftrightarrow \mathcal{L}(\sigma) \geq \mathcal{J}(\sigma), \quad \forall\, \sigma \in \Xi$$
$$\mathcal{L} \succ \mathcal{J} \Leftrightarrow \mathcal{L}(\sigma) > \mathcal{J}(\sigma), \quad \forall\, \sigma \in \text{dom}(\mathcal{J})$$

For any $u, v \in \mathbb{R}^n$, we also note:

$$u \succeq v \Leftrightarrow u_k \geq v_k, \quad \forall\, k \in [n]$$
$$u \succ v \Leftrightarrow u_k > v_k, \quad \forall\, k \in [n].$$

Definition 1.2 (descending rank): Let $v \in \mathbb{R}^n$ and let $\rho : [n] \to [n]$ be a bijection such that the vector $\hat{v} = (v_{\rho^{(-1)}(0)}, \ldots, v_{\rho^{(-1)}(n-1)})^T$ is sorted in *descending* mode.

For any $k \in [n]$, we call $\rho(k)$ the *descending rank* of v_k (with respect to ρ).

Remark 1.1: It comes that if $\rho(k) = j$, then v_k is, according to the sorting induced by ρ, the $(j+1)$th biggest element of v.

2. Tunable (inexact) oracles

Here below we aim at defining the class of Tunable Oracles Methods (TOM) from which one can benefit by adapting the amount of inexactness in the oracles involved at each iteration. Prior to this goal, we recall some findings about methods incorporating inexactness and we introduce our assumed oracle cost model. Finally, we substantiate the concept of Tunable Oracles Methods with thee complete examples serving as common thread.

2.1. Impact of inexact oracles

The worst-case behaviour analysis of iterative methods relying on inexact oracles displays some favourable structure. Let $N \in \mathbb{N}$ be the number of performed iterations, one defines

the total amount of inexactness at iteration $k \in \{0, \dots, N-1\}$ as δ_k. Researchers come up with convergence models

$$\mathcal{C}(N) \leq \mathcal{E}(\{\delta_k\}_{k=0}^{N-1}) \tag{5}$$

where

- $\mathcal{C}(N)$ acts as a positive *gauge* one aims to minimize
- $\mathcal{E} : \mathbb{R}_+^N \to \mathbb{R}_+$ informs about convergence under the impact of a sequence $\{\delta_k\}_{k=0}^{N-1}$.

$\mathcal{C}$ gauges include gradient mapping norms [5], functional gaps [3,21].
As in Example 1.1 inequality (4), a possible instance for our model could show up as

$$\underbrace{F(\hat{x}_N) - F^*}_{\mathcal{C}(N)} \leq \underbrace{\frac{M(N) + 3\sum_{k=0}^{N-1}(1-\kappa)^{N-1-k}\,\delta_k}{\sum_{k=0}^{N-1}(1-\kappa)^{N-1-k}}}_{\mathcal{E}(\{\delta_k\}_{k=0}^{N-1})}$$

Remark 2.1: Within the *error-free* framework, i.e. $\delta_k = 0$ for every integer k,

$$\lim_{N\to\infty} \mathcal{E}(\{0\}_{k=0}^{N-1}) = 0.$$

When accounting for errors, one theoretically observes convergence only up to $\mathcal{E}(\{\delta_k\}_{k=0}^{N-1})$ accuracy. It can happen that the schedule $\{\delta_k\}_{k=0}^{N-1}$ does not decrease sufficiently fast leading to a well-known phenomenon referred to as *error-accumulation* in the literature [2]. This latter translates to

$$\lim_{N\to\infty} \mathcal{E}(\{\delta_k\}_{k=0}^{N-1}) = \infty.$$

2.2. Cost of inexact oracles

Without loss of generality, one can define a *reference* inexactness $\bar{\delta} > 0$ together with constants $0 \leq m < 1 < M < \infty$ such that the inexactness of the oracles fall in the segment $\Xi := [m\,\bar{\delta}, M\,\bar{\delta}]$. This means that at each iteration $k \in \{0, \dots, N-1\}$, one allows the user to freely pick up any $\delta_k \in \Xi$. As previously unveiled, such request costs a computational tribute, namely $\mathcal{B}_k(\delta_k) \geq 0$. Suitable for a variety of applications [5,7,13], we suggest the oracle cost model

$$\mathcal{B}_k(\delta_k) = b_k\, h(\delta_k), \tag{6}$$

where

- $b_k > 0$ denotes the cost distortion of iteration k
- $h : \Xi \to \mathbb{R}_+$ dictates how the cost fluctuates with δ_k.

Recalling once again the introduction, one identifies the *a priori* number ω_k of inner iterations to obtain δ_k inexact information about F as a multiple of

$$\mathcal{B}_k(\delta_k) = \underbrace{\sqrt{\kappa(G)^{-1}}}_{b_k}\ \underbrace{\log(\delta_k^{-1})}_{h(\delta_k)}.$$

Remark 2.2: When unknown or when no argument justifies that any iteration turns out to be less expensive than another, we arbitrarily set $b_k = 1$ for any integer $k \geq 0$. Let us also point out that in the absence exogenous indication, $m = 0$ and $M = \infty$.

2.3. Tunable oracle method

An instance of Tunable Oracle Methods (TOM) stands as an N-step *iterative* algorithm $\mathcal{A}$ whose iterations involve a notion of *controllable* δ-inexactness as in (6) such that $\mathcal{A}$ converges in the sense of (5). Furthermore, to truly exploit such *controllable* feature, we require the explicit knowledge of $\{b_k\}_{k=0}^{N-1}$, $\mathcal{E}$ and h, up to a multiplicative constant factor. Such desire emphasizes the *offline* nature focus of the paper at this stage. We make use of *a priori* information about the behaviour of $\mathcal{A}$ and the expected cost one should encounter while requesting a schedule of $\{\delta_k\}_{k=0}^{N-1}$ inexact oracles. We explain later on how to take advantage of FOM in an *online* setting.

2.4. Guideline examples

Throughout the sequel, we make use of the following:

- $\Psi : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ proper closed convex function, not necessarily smooth.
- $\ell : \mathbb{R}^q \to \mathbb{R}$ closed convex function, $L_\ell < \infty$ Lipschitz continuous and simple, i.e. its proximal operator can be computed at negligible cost .[1]
- $c : \mathbb{R}^d \to \mathbb{R}^q$ a smooth map with $L_{\nabla c} < \infty$ Lipschitz continuous Jacobian ∇c.

Combining these ingredients, we finally introduce $f = \ell \circ c$ and $F = f + \Psi$ nonconvex, nonsmooth in the general case. We will assume that

$$< /p >< p > F^* = \min_{x \in \mathbb{R}^d} F(x) > -\infty.$$

Example 2.1 (Composite convex optimization with inexact proximal operator): The inexact Accelerated Forward–Backward algorithm (iAFB) from [13] tackles so-called convex composite problems, ubiquitous in image processing [1]. The specificities of this class read: $q = 1$, $\ell = \mathrm{id}_{\mathbb{R}}$, convexity (respectively $\mu \geq 0$ strong-convexity) of c (respectively Ψ) thus $f = c$ and ∇f is $L_f = L_{\nabla c}$ Lipschitz continuous. Among other appealing features, iAFB presented therein allows for Ψ whose proximal mapping is not simple, e.g. in *sparse overlapping groups regularization* [19]. At $z \in \mathbb{R}^d$, $\lambda > 0$, the primal–dual pair of problems related to the proximal step of a function ϕ translates to

$$\min_{x \in \mathbb{R}^d} \left\{ \Phi_p(x; \lambda, z, \phi) := \frac{1}{2} ||x - z||^2 + \lambda \phi(x) \right\} \tag{P}$$

$$\max_{v \in \mathbb{R}^d} \left\{ \Phi_d(v; \lambda, z, \phi) := \frac{1}{2} \left(||z||^2 - ||z - \lambda v||^2 \right) - \lambda \phi^*(v) \right\}, \tag{D}$$

where ϕ^* is the Fenchel conjugate of ϕ. iAFB produces iterates x_k, y_k, z_k, v_k thanks to coefficients $\lambda_k \in \mathcal{O}(L_f)$ and $A_k \in \Omega(k^2)$ for any integer $k \geq 0$. Either iAFB employs predefined sequence of stepsizes $\{\lambda_k^{-1}\}_{k \in \mathbb{N}}$ and then A_k, accounting as certificate sequence, can be

computed in advance, i.e. *offline*, or it adopts Armijo line search to adapt to local smoothness λ_k and one only has access to A_k for iterations $k' \geq k$, in an *online* fashion. Following authors' notations, (x_{k+1}, v_{k+1}) stems as a δ_k inexact output from the proximal step oracle at iteration $k \geq 0$ if

$$\mathrm{PD}_{\lambda\phi}(x, v; z) := \Phi_p(x; \lambda, z, \phi) - \Phi_d(v; \lambda, z, \phi) \leq \delta_k \tag{7}$$

with $x = x_{k+1}$, $v = v_{k+1} - \mu\, x_{k+1}$, $\lambda = \frac{\lambda_k}{1+\lambda_k\mu}$, $\phi = \Psi - \frac{\mu}{2}||\cdot||^2$ and $z = \frac{y_k - \lambda_k \nabla f(y_k)}{1+\lambda_k\mu}$.

Under this notion of inexactness, after $N \in \mathbb{N}$ steps, the following guarantees hold:

$$\underbrace{F(x_N) - F^*}_{\mathcal{C}(N)} \leq \underbrace{\frac{4R^2 + \sum_{k=0}^{N-1} (A_{k+1} \cdot (1+\mu\lambda_k)^2 \cdot \lambda_k^{-1})\, \delta_k}{A_N}}_{\mathcal{E}\left(\{\delta_k\}_{k=0}^{N-1}\right)}, \tag{8}$$

where $R < \infty$ denotes the distance from x_0 to the set of global minimizers of F.

The presumed oracle cost, i.e. the work $\mathcal{B}_k(\delta_k)$ needed to produce (x_{k+1}, v_{k+1}) fulfilling (7) remains problem dependent. It is also highly influenced by the technique employed to solve the pair (P) / (D). In [13], authors deal with CUR factorization problem with *sparse overlapping groups regularization*. They simply use FISTA, a *fast* first-order method, to solve (D) yielding a linked sequence of primal recovered iterates (P). In the present setting, it is shown in [11] that the primal–dual gap decreases sublinearly as ω_k^{-1} where ω_k denotes the number of steps of FISTA in (D). Then, for an arbitrarily chosen $b_k = 1$ (see Remark 2.2), we model the cost as

$$\mathcal{B}_k(\delta_k) = \underbrace{\delta_k^{-1}}_{h(\delta_k)}. \tag{9}$$

Example 2.2 (Composition of convex function optimization): The paper [5] considers the most general setting for which global optimality is, obviously, out of reach. Let us focus on a single method analysed therein, the inexact Prox-Linear algorithm, iPL in short terms. Given $z \in \mathrm{dom}(\Psi)$ and $t > 0$, one defines the functional

$$F_t(\cdot; z) = \Psi(\cdot) + \ell(c(z) + \nabla c(z)(\cdot - z)) + \frac{t^{-1}}{2}||\cdot - z||^2.$$

We call δ inexact solution (x_+, ξ) of the minimization of $F_t(\cdot\,; z)$ a pair fulfilling the properties: $||\xi|| \leq \delta$ and

$$x_+ \in \arg\min_{x \in \mathbb{R}^d} \Psi(\cdot) + \ell(\xi + c(z) + \nabla c(z)(\cdot - z)) + \frac{t^{-1}}{2}||\cdot - z||^2.$$

In the case of an exact $(x_+, 0)$ (i.e. $\delta = 0$ inexact) solution, the proximal gradient mapping $\mathcal{G}_t(x)$ at z with stepsize $t \leq (L_\ell\, L_{\nabla c})^{-1}$ stands as the vector $\mathcal{G}_t(x) = t^{-1}(x - x_+)$ if and only if $F(x_+) \leq F_t(x_+; z)$. Its norm constitutes a measure of stationarity [5] that generalizes the gradient norm in the smooth unconstrained optimization framework. When calling up

$\{\delta_k\}_{k=0}^{N-1}$ inexact solutions at successive iterates $z = x_k \in \mathrm{dom}(\Psi)$ and stepsizes $t = t_k \in]0, (L_\ell\, L_{\nabla c})^{-1}]$, iPL produces iterates $\{x_k\}_{k\geq 0}$ such that:

$$\underbrace{\min_{k'\in\{1,\ldots,N\}} ||\mathcal{G}_{t_k}(x_{k'})||^2}_{\mathcal{C}(N)} \leq \underbrace{\frac{\sum_{k=0}^{N-1} 4\, t_k^{-1}\left(F(x_k) - F(x_{k+1}) + 4\, L_\ell\, \delta_k\right)}{N}}_{\mathcal{E}\left(\{\delta_k\}_{k=0}^{N-1}\right)}. \tag{10}$$

Again, the oracle cost should be derived from the complexity of the inner method used to obtain the sequence of $\{\delta_k\}_{k=0}^{N-1}$ inexact solutions at pairs $\{(x_k, t_k)\}_{k=0}^{N-1}$. When no further specific structure is taken as granted, excepted an easy proximal mapping of Ψ (unlike the previous example (8)), [5] suggests to exploit duality. At any iteration $k \geq 0$, one can compute a δ_k inexact pair (x_{k+1}, ξ_{k+1}) as previously explained by finding a subgradient ξ_{k+1} of the (negated) Fenchel conjugate function of $F_{t_k}(\cdot\,; x_k)$ whose norm does not exceed δ_k. With the oracles at hand and accounting for the fact that this conjugate boils down to a sum of a smooth convex term and a proximable nonsmooth convex function, A-HPE from [15] ensures a minimal subgradient norm of order $\mathcal{O}(t_k||\nabla c(x_k)||^2\, \omega_k^{-\frac{3}{2}})$ after ω_k A-HPE iterations.

Hence, it comes that the number of A-HPE iterations required to get a δ_k inexact pair scales as $t_k^{\frac{2}{3}}\, ||\nabla c(x_k)||^{\frac{4}{3}}\, \delta_k^{-\frac{2}{3}}$. Obviously, one does not know in advance the value $||\nabla c(x_k)||$ since it relies on x_k, only computable from iteration $k \geq 0$. Remark 2.2 then suggests that any $\gamma > 0$, e.g. $\gamma = 1$, can serve as artificial upper bound so that either iPL uses predefined stepsizes $\{t_k\}_{k\in\mathbb{N}}$ (*offline*) or adjusted stepsizes (*online*).

$$\mathcal{B}_k(\delta_k) = \underbrace{t_k^{\frac{2}{3}}}_{b_k}\, \underbrace{\delta_k^{-\frac{2}{3}}}_{h(\delta_k)}. \tag{11}$$

Note that if $\mathrm{dom}(\Psi)$ is bounded with known diameter, one can use regularization and apply Nesterov fast gradient method (FGM) on strongly-convex composite objectives to obtain an enhanced $t_k^{\frac{1}{2}}\, ||\nabla c(x_k)||\, \delta_k^{-\frac{1}{2}}$ complexity [17]. We emphasize that *stopping criteria* are readily available to check whether δ_k accuracy has been reached.

Example 2.3 (Robust optimization on convex hull): Closely related to [13,21] extends the analysis of FGM under the presence of inexactness by allowing f to be smooth and convex relative to some Legendre kernel function [11], covering the so-called Bregman setting. Therefore, their results also encompass the Euclidean setting which we stick to for the sake of simplicity. We assume that Ψ represents an indicator function of a convex subset $X \subseteq \mathbb{R}^d, q = 1, \ell = \mathrm{id}_{\mathbb{R}}$ and c is $\mu \geq 0$ strongly convex. That is, f is $\mu \geq 0$ strongly convex and $\nabla f = \nabla c$ is $L_f = L_{\nabla c}$ Lipschitz continuous. Let $\Theta = \{\theta_i\}_{i=1}^n$ be a collection of $n \in \mathbb{N}$ vectors from $\mathbb{R}^d$ dubbed as *scenarios* and let $\sigma > 0$. In *robust optimization*, one might be interested in minimizing the function

$$f : X \to \mathbb{R}, \quad x \to f(x) = \frac{\mu}{2}\, ||x||^2 + \max_{\theta\in\mathbf{conv}(\Theta)} \langle\theta, x\rangle - \frac{\sigma}{2}||\theta - \bar{\theta}||^2$$

for some *anchor scenario* $\bar{\theta}$, e.g. $\bar{\theta} = n^{-1}\, \sum_{i=1}^n\, \theta_i$.

In other words, one would like to minimize a (regularized) linear objective taking into account that the cost vector could be any convex combination of previously encountered costs $\{\theta_i\}_{i=1}^n$. Akin to Example 1.1, we deduce that the *exact* gradient of f at any $x \in X$ reads $\nabla f(x) = \mu x + \theta^*(x)$ where $\theta^*(x) = \arg\max_{\theta \in \mathbf{conv}(\Theta)} \langle \theta, x\rangle - \frac{\sigma}{2}||\theta - \bar{\theta}||^2$ and gradient's Lipschitz constant $L_f = \sigma^{-1}$.

An approximate maximizer $\theta_x \in \mathbf{conv}(\Theta)$ of the problem defining f at x that verifies

$$\langle (\theta^*(x) - \theta_x), x\rangle + \frac{\sigma}{2}\left(||\theta_x - \bar{\theta}||^2 - ||\theta^*(x) - \bar{\theta}||^2\right) \leq \delta \tag{12}$$

can be used to construct $\tilde{f}(x) = \frac{\mu}{2}||x||^2 + \langle \theta_x, x\rangle - \frac{\sigma}{2}||\theta_x - \bar{\theta}||^2 \simeq f(x)$ and $\nabla\tilde{f}(x) = \mu x + \theta_x \simeq \nabla f(x)$, providing $(2\delta, 2L_f + \mu, \mu)$ inexact information as originally understood in [2]. [21] show that FGM ([21], Algorithm 2) involving a sequence of $\{\delta_k\}_{k=0}^{N-1}$ inexact information and fixed stepsizes produces a final iterate x_N such that

$$\underbrace{F(x_N) - F^*}_{\mathcal{C}(N)} \leq \underbrace{\frac{R^2 + 2\sum_{k=0}^{N-1} A_{k+1}\,\delta_k}{A_N}}_{\mathcal{E}\left(\{\delta_k\}_{k=0}^{N-1}\right)}, \tag{13}$$

where, once again, $R < \infty$ denotes the distance from x_0 to the set of global minimizers of F. As for iAFB, the coefficients A_k for $k \geq 0$ serve as convergence certificates and are involved in subtle convex combinations of iterates within FGM. In the present setting, A_k's value can be determined beforehand, $A_k \in O(\max\{k^2, (1 + \frac{1}{4}\sqrt{\mu/(\mu + 2L_f)})^{2k}\})$.

Let $O = [\theta_1, \ldots, \theta_n]^T$ and let $\hat{\kappa} := \frac{\lambda_{\min}(OO^T)}{\lambda_{\max}(OO^T)}$. By using another version of FISTA described in [23], one can take advantage of possible strong convexity, i.e. $\hat{\kappa} > 0$, of the usual reformulation of the inner problem :

$$\max_{\theta \in \mathbf{conv}(\Theta)} \langle \theta, x\rangle - \frac{\sigma}{2}||\theta - \bar{\theta}||^2 = \max_{w \geq \mathbf{0},\, ||w||_1 = 1} \langle O^T w, x\rangle - \frac{\sigma}{2}||O^T w - \bar{\theta}||^2.$$

Assuming that $\hat{\kappa} > 0$, just as in Example 1.1, one can link the work ω_k to obtain δ_k accurate θ_{x_k} at iteration $k \geq 0$ by writing

$$\mathcal{B}_k(\delta_k) = \underbrace{\sqrt{\hat{\kappa}^{-1}}}_{b_k}\,\underbrace{\log(\delta_k^{-1})}_{h(\delta_k)}, \tag{14}$$

whereas in the absence of strong convexity, i.e. $\hat{\kappa} = 0$, one would rather set $\mathcal{B}_k(\delta_k) = \sqrt{\lambda_{\max}(OO^T)}\,\delta_k^{-\frac{1}{2}}$ as common for smooth convex optimization [17]. We explain in Section 4 how one can easily monitor the quality of an approximate solution for (12).

Remark 2.3: Usually, the workload $\omega \propto \mathcal{B}(\delta)$ to obtain (the output) of δ inexact oracles stays *by nature* an integer quantity, e.g. a number of inner iterations (Examples 2.1, 2.2 and 2.3). For the sake of simplicity, however, we will accept that it varies continuously as in the formulas (9), (11), (14).

3. Optimal inexactness schedules

Now that the impact and the cost of inexact oracles have been introduced, we can elaborate about the main objective of this paper. We assume for the time being that $N \in \mathbb{N}$ is fixed. One can try to suffer the smallest possible effect from inexact oracles to ensure the best worst-case convergence upper-bound. Obviously, one would like each δ_k to match its best value $m\bar{\delta}$. However, sometimes one cannot afford a schedule $\{\delta_k = m\bar{\delta}\}_{k=0}^{N-1}$ if one limits the overall computational budget. On the other hand, when not obliged to, it is not advisable to ask for the worst oracle accuracy at each iteration, i.e. $M\bar{\delta}$, yielding the inexactness schedule $\{\delta_k = M\bar{\delta}\}_{k=0}^{N-1}$.

Thereby, we propose to solve a master problem that aims at devising the optimal *trade-off* between the costs of oracles and worst-case guarantee harms due to their inexactness. We start by providing our blanket Assumptions A and B.

3.1. Framework

From now on we will make a little abuse of notation about the *reference* inexactness $\bar{\delta} > 0$. Depending on the context, this latter will either depict a real value, an N-step inexactness schedule $\{\delta_k = \bar{\delta}\}_{k=0}^{N-1}$ or even the corresponding vector in $\mathbb{R}_+^N$, i.e. $\bar{\delta}\,\mathbf{e}$.

Assumption A: We have access to $a \succ \mathbf{0}$, $\exists\, \tilde{\mathcal{E}} : \mathbb{R}_+ \to \mathbb{R}_+$ *strictly increasing* with

$$\mathcal{E}(\delta) := \mathcal{E}(\{\delta_k\}_{k=0}^{N-1}) = \tilde{\mathcal{E}}\left(\sum_{k=0}^{N-1} a_k\,\delta_k\right) \tag{15}$$

We now invoke Assumption B suggesting that one is able to predict the overall cost $\mathcal{B}(\delta)$ of a schedule of inexactness δ. Furthermore, some technicalities about the structure of the function h from (6) are stated.

Assumption B: We have access to $b \succ \mathbf{0}$ and $h : \Xi \to \mathbb{R}_+$ *differentiable*, *invertible*, *strictly decreasing* with

$$\mathcal{B}(\delta) := \sum_{k=0}^{N-1} \mathcal{B}_k(\delta_k) = \sum_{k=0}^{N-1} b_k\,h(\delta_k) \tag{16}$$

$h' : \mathrm{dom}(h') \supseteq \mathrm{int}(\Xi) \to \mathrm{Im}(h') \subseteq \mathbb{R}_-$ must be *invertible* and *strictly increasing*.

Remark 3.1: Assumption A tells us that on any subset of $\mathbb{R}_+^N$, minimizing $\mathcal{E}(\{\delta_k\}_{k=0}^{N-1})$ boils down to minimizing $\sum_{k=0}^{N-1} a_k\,\delta_k$, i.e. a minimizer of $a^T\,\delta$ stays optimal for $\mathcal{E}(\delta)$.

Remark 3.2: In what concerns our examples, one can identify problem-dependent constants $C_1, C_2 > 0$ such that $\tilde{\mathcal{E}}$ admits a shared structure $s \to \tilde{\mathcal{E}}(s) = C_1 + C_2\,s$.

In their adaptive *online* versions, sometimes practically more attractive, parameters λ_k (iAFB) and t_k (iPL) are not known in advance since they rely on line searches at iteration $k \geq 0$. Unfortunately, they influence the values of a_k/b_k's. Thenceforth, Examples 2.1 and 2.2 would fail to satisfy Assumption A and/or B. Nevertheless, as already

argued, one can fix $\lambda_k \leq L_{\nabla c}^{-1}$ (iAFB), $t_k^{-1} \geq L_\ell \cdot L_{\nabla c}$ (iPL) for every $k \geq 0$ and avoid the line searches. In such *offline* circumstances, both $\{a_k\}_{k\geq 0}$ and $\{b_k\}_{k\geq 0}$ become accessible and the assumptions are fulfilled. If taken constants, i.e. for every $k \geq 0$, $\lambda_k = \lambda$, $t_k = t$ for well-chosen $\lambda, t \in \mathbb{R}_+$, then $A_k \in \mathcal{O}(\max\{k^2, (1 - \sqrt{\mu/L_f})^{-k}\})$ for iAFB, $A_k \in \mathcal{O}(\max\{k^2, (1 + \frac{1}{4}\sqrt{\mu/(\mu + 2L_f)})^{2k}\})$ in FGD and $t_k \in \mathcal{O}(1)$ in iPL.

3.2. Master problems

Accuracy controlled

In its most standard version, we design a master problem $\delta^*(N, \bar{\delta}, m, M)$ for which the degrees of freedom reside in the accuracies of the N iterations of our TOM, i.e. $\delta \in \mathbb{R}_+^N$. Given a *reference* $\bar{\delta} \in \mathbb{R}_+$ precision, we implicitly deduce the total allocated computational budget as $\sum_{k=0}^{N-1} \mathcal{B}_k(\bar{\delta}) = (\sum_{k=0}^{N-1} b_k)\, h(\bar{\delta})$.

Controlling $\delta \in [m\bar{\delta}, M\bar{\delta}]^N$, we minimize the bound $\mathcal{E}(\delta)$ under the budget constraint $\mathcal{B}(\delta) = \mathcal{B}(\bar{\delta})$. Taking into account Assumptions A, B and Remark 3.1, one translates

$$\delta^*(N, \bar{\delta}, m, M) \in \arg \min_{m\bar{\delta} \preceq \delta \preceq M\bar{\delta}} \sum_{k=0}^{N-1} a_k \delta_k \quad \text{s.t.} \sum_{k=0}^{N-1} b_k\, h(\delta_k) = \mathcal{B}(\bar{\delta}). \tag{17}$$

3.2.1. General solutions

We present a first theorem which will proved to be useful to guarantee some consistency in the optimal schedules. Its scope encompasses more general structured *non-linear* resource allocation problems. Related results can be found deeply rooted in the literature, the reader should refer to [8] and the references therein.

Theorem 3.1: *Let $N \in \mathbb{N}$, $\Xi = [l, u] \subseteq \mathbb{R}$ and $D > 0$. Let $\{n_k\}_{k=0}^{N-1}$ and $\{d_k\}_{k=0}^{N-1}$ be two sequences of continuously differentiable functions on Ξ and $\{n_k\}_{k=0}^{N-1}$ elements are furthermore strictly increasing on Ξ. If for every $k \in [N]$, the ratio $\mathcal{I}_k = n_k'/d_k'$ is strictly increasing and strictly negative on Ξ then any solution σ^* of*

$$\min_{\sigma \in \Xi^N} \sum_{k=0}^{N-1} n_k(\sigma_k) \quad \text{s.t.} \sum_{k=0}^{N-1} d_k(\sigma_k) = D \tag{18}$$

admits as a property that for all pairs of indices $(k_1, k_2) \in [N]^2$,

$$\mathcal{I}_{k_1} \succeq \mathcal{I}_{k_2} \Rightarrow \sigma_{k_1}^* \geq \sigma_{k_2}^* \tag{19}$$

Moreover,

$$\mathcal{I}_{k_1} \succ \mathcal{I}_{k_2} \wedge \sigma_{k_1}^*, \sigma_{k_2}^* \in]l, u[\Rightarrow \sigma_{k_1}^* > \sigma_{k_2}^*. \tag{20}$$

Proof: We prove Theorem 3.1 in Appendix A.1. ■

Example 3.2: The reader can get more insight thanks to the following example.

For every $k \in [N]$, let $\sigma \to n_k(\sigma) = (k+1)^3\,\sigma$, $\sigma \to n_k(\sigma) = -\log(\sigma)$, $l = 1$ and $u = 10^5$. The functional ratios $\mathcal{I}_k$ are given by

$$\mathcal{I}_k :]l, u[\to \mathbb{R}, \quad \sigma \to \mathcal{I}_k(\sigma) = \frac{n_k'(\sigma)}{d_k'(\sigma)} = \frac{(k+1)^3}{-\sigma^{-1}} = -(k+1)^3\,\sigma^{-1}.$$

On $\Xi = [l, u]$, n_k's and d_k's are continuously differentiable and n_k's are increasing while $\mathcal{I}_k$'s are strictly increasing and strictly negative. In addition, if $k_1 \leq k_2$,

$$\mathcal{I}_{k_1} \succeq \mathcal{I}_{k_2}$$

since for every $\sigma \in \Xi, \mathcal{I}_{k_1}(\sigma) = -(k_1+1)^3\,\sigma^{-1} \geq -(k_2+1)^3\,\sigma^{-1} = \mathcal{I}_{k_2}(\sigma)$.

Reordering In the case where all the ratios $\{\mathcal{I}_k\}_{k=0}^{N-1}$ can be ordered, i.e. there exists a bijective mapping $\tau : [N] \to [N]$ such that for all pair of indices $(k_1, k_2) \in [N]^2$,

$$\tau(k_1) < \tau(k_2) \Rightarrow \mathcal{I}_{k_1} \succeq \mathcal{I}_{k_2}. \tag{21}$$

Theorem 3.1 suggests that the entries $\{\sigma_k^*\}_{k=0}^{N-1}$ of an optimal solution σ^* of (18) can be *ranked* as well. Provided a suitable comparison vector $\nu \in \mathbb{R}^N$, we advocate the usefulness of ρ, *descending rank* function based on ν to act like a τ function above, i.e. for any $k \in [N]$, we would have

$$\rho(k) = \tau(k). \tag{22}$$

As a reminder from Definition 1.2, we write $\rho(k) = j$ if ν_k is the $(j+1)$th largest element of ν. It follows from Theorem 3.1 that σ_k^* must correspond to the $(j+1)$th largest element of σ^*. As displayed in Theorem 3.3, 3.1 applies verbatim to problem (17) by picking for every $k \in [N]$, $n_k(\sigma_k) = a_k\,\sigma_k$, $d_k(\sigma_k) = b_k h(\sigma_k)$ and $D = \mathcal{B}(\bar{\delta})$. One valid comparison vector that fulfils (21) and (22) would be $\nu_k = b_k/a_k$.

Let δ^* be an optimal solution for (17). We summarize last paragraph's key content by underlining the fact that the biggest the value of b_k/a_k will be, the biggest the optimal inexactness at iteration $k \in [N]$, according to our master problem, will be as well.

$$\rho(k_1) < \rho(k_2) \Rightarrow \nu_{k_1} = \frac{b_{k_1}}{a_{k_1}} \geq \frac{b_{k_2}}{a_{k_2}} = \nu_{k_2} \Rightarrow \delta_{k_1}^* \geq \delta_{k_2}^*.$$

We are now ready to state a general theorem about master problem (17).

Theorem 3.3: *Let Assumptions* A *and* B *hold with* $(a, b) \in \mathbb{R}_{++}^{N\times 2}$, $m < 1 < M$ *and* $h : [m\bar{\delta}, M\bar{\delta}] \to \mathbb{R}_+$ *being convex.* $\exists N_\oplus, N_\ominus \in \{0, \ldots, N-1\}$, $\lambda^* \in \mathbb{R}$ *such that* $\forall\ k \in$

$\{0,\dots,N-1\}$,

$$\delta_k^* = \begin{cases} M\,\bar{\delta} & k \in \oplus := \{\tilde{k} \,|\, \rho(\tilde{k}) < N_\oplus\} \\ (h')^{(-1)}\left(\frac{a_k\,\lambda^*}{b_k}\right) & k \in \mathcal{T} := \{\tilde{k} \,|\, N_\oplus \le \rho(\tilde{k}) \le N-1-N_\ominus\} \\ m\,\bar{\delta} & k \in \ominus := \{\tilde{k} \,|\, \rho(\tilde{k}) > N-1-N_\ominus\}, \end{cases} \tag{23}$$

where $\rho(k)$ depicts the descending rank of $v_k = b_k/a_k$, λ^* satisfies the equality

$$\sum_{k=0}^{N-1} b_k\,h(\bar{\delta}) - \left[h(M\bar{\delta})\left(\sum_{k\in\oplus} b_k\right) + h(m\bar{\delta})\left(\sum_{k\in\ominus} b_k\right)\right] = \sum_{k\in\mathcal{T}} b_k\,h\left((h')^{(-1)}\left(\frac{a_k\,\lambda^*}{b_k}\right)\right) \tag{24}$$

and δ^* stands as a solution of (17).

Proof: The proof of Theorem 3.3 is given in Appendix A.2. ■

Remark 3.3: If all the entries of $\nu \in \mathbb{R}^N_{++}$ differ, their ordering is unique, so becomes δ^* as shown in Appendix A.3.

Remark 3.4: One only need to know a_k (respectively b_k) up to a common factor $K_a > 0$ (respectively $K_b > 0$). That is, it is sufficient to know $\tilde{a}_k$ (respectively $\tilde{b}_k$) such that for any $k \in [N]$, $a_k = K_a\,\tilde{a}_k$ (respectively $b_k = K_b\,\tilde{b}_k$). In Theorem 3.3, instead of looking for λ^*, one then searches for another constant $\tilde{\lambda}^* = \lambda^*\,K_a/K_b$ that would act like $\tilde{\lambda}^*\,\tilde{a}_k/\tilde{b}_k = \lambda^*\,a_k/b_k$.

Example 3.4 (Toy example): We want here to give a first glimpse about the upcoming optimal schedules that will apply for (among others) situations reflected in Examples 2.1, 2.2 and 2.3. To this purpose, we clarify the above notations by writing explicitly what they entail given an academic toy example rightfully meeting Assumptions A and B. We consider that the *impact coefficients* and *relative costs* of oracles are given for any $k \in \{0,\dots,N-1=79\}$ by

$$a_k = (k+1) \quad b_k = \begin{cases} \frac{3}{420} & 0 \le k < 20 \\ \frac{2}{420} & 20 \le k < 40 \\ \frac{8}{420} & 40 \le k < 80. \end{cases}$$

The *reference* inexactness parameter is chosen as $\bar{\delta} = 10^{-4}$ and $m = 0 < 1 < M = 2$.

The oracle cost $h : [0,1] \to [0,\infty)$ is convex, differentiable and fluctuates polylogarithmically with $0 \le \delta \le 1$ as $\delta \to h(\delta) = \log^2(\delta^{-1})$. h' is negative and strictly increasing from $[0,1]$ to $(-\infty,0]$, its inverse $(h')^{(-1)} : (-\infty,0] \to [0,1]$ exists for $-\omega < 0$

$$-\omega \to (h')^{(-1)}(-\omega) = 2\,\omega^{-1}\,\mathcal{W}_0(\omega/2),$$

where $\mathcal{W}_0$ depicts the Lambert $\mathcal{W}$ function on its 0-branch. Here, Theorem 3.3 applies. Obviously, $\lim_{\delta\to 0} h(\delta) = \infty$ translates to $\ominus = \emptyset \Leftrightarrow N_\ominus = 0$. From that point, we can efficiently solve KKT conditions. They inform that for our present problem instance, $N_\oplus = 10$. Fortunately, the indices $k \in [80]$ for which v_k values are the $N_\oplus$ biggest fall in $\oplus = \{0,\dots,9\}$. We can conclude that the set $\mathcal{T}$ contains the indices $\{10,\dots,79\}$.

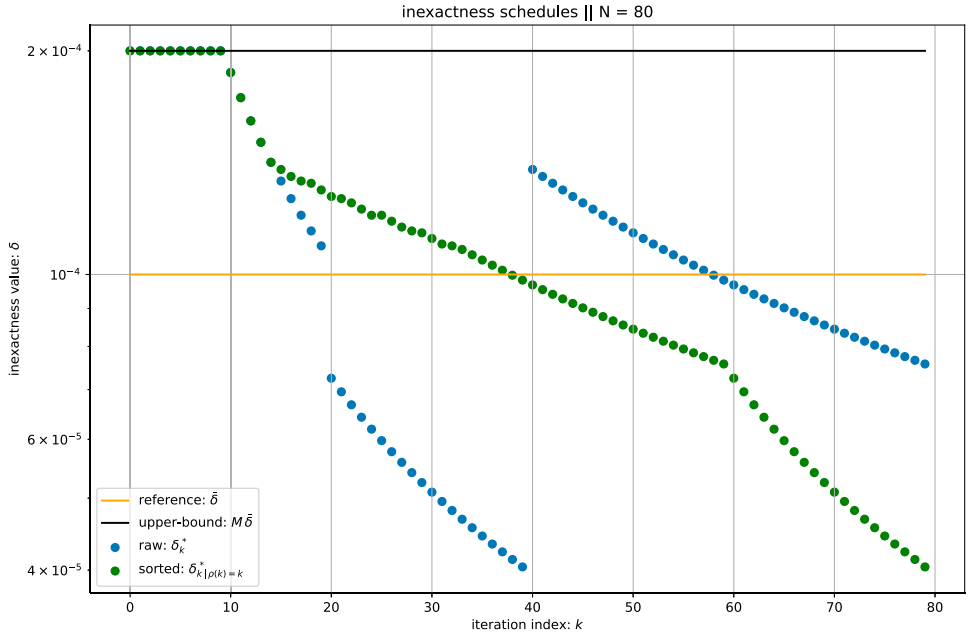


Figure 1. Inexactness schedules: *reference*, raw optimal $\{\delta_k^*\}_{k=0}^{N-1}$ and sorted optimal $\{\delta^*_{\tilde{k}\,|\,\rho(\tilde{k})=k}\}_{k=0}^{N-1}$.

We summarize the calculated optimal schedules for (17) in (25) bearing in mind that $\lambda = -\lambda^* \simeq 27.5757$. Figure 1 graphs the optimal inexactness schedule in (25).

$$\delta_k^* = \begin{cases} 2 \cdot 10^{-4} & k \in \{0, \dots, 9\} \\ \left(\frac{2\,(3/420)}{(k+1)\,\lambda}\right) \mathcal{W}_0\left(\frac{(k+1)\,\lambda}{2\,(3/420)}\right) & k \in \{10, \dots, 19\} \\ \left(\frac{2\,(2/420)}{(k+1)\,\lambda}\right) \mathcal{W}_0\left(\frac{(k+1)\,\lambda}{2\,(2/420)}\right) & k \in \{20, \dots, 39\} \\ \left(\frac{2\,(8/420)}{(k+1)\,\lambda}\right) \mathcal{W}_0\left(\frac{(k+1)\,\lambda}{2\,(8/420)}\right) & k \in \{40, \dots, 79\}. \end{cases} \tag{25}$$

3.2.2. Closed-form solutions

In this section, we focus on analytical closed-form solutions we can obtain from the previous theorems when specifying a certain type of h function, related to the practical examples motivating our oracle cost model (6). Yet, it usually remains to seize the correct values of $N_\oplus$ (number of iterations either performed at worst oracle accuracy or involving the least computational efforts) and $N_\ominus$ (number of iterations either achieved with the best oracle accuracy or demanding the heaviest computational efforts).

To circumvent a cautious search for the right pair $(N_\oplus, N_\ominus) \in \{0, \dots, N-1\}^2$, one can immediately detect whether $N_\oplus = 0 = N_\ominus$ with a simple trial in constant time. It is essentially what we achieve in Corollaries 3.5 and 3.6. In such circumstances, $\oplus = \emptyset = \ominus$

(Theorem 3.3), i.e. the *transient set* $\mathcal{T}$ that normally collects the k-indices of iterations associated with ν_k values smaller than the $N_\oplus$ biggest and bigger than the $N_\ominus$ smallest involves here all the iterations, i.e. $\mathcal{T} = \{0, \dots, N-1\}$.

Functional family Let us formally declare a meaningful functional family of h functions that captures our applications of interest. For any $r > 0$, we define the convex function $h_r : \mathbb{R}_+ \to \mathbb{R}_{++}$ and the inverse of its derivative $(h_r')^{(-1)} : \mathbb{R}_{--} \to \mathbb{R}_+$

$$h_r(\delta) = \delta^{-r} \Rightarrow (h_r')^{(-1)}(-\omega) = \left(\frac{\omega}{r}\right)^{-\frac{1}{r+1}}. \tag{26}$$

Corollary 3.5: *Let Assumptions* A *and* B *hold with* $(a, b) \in \mathbb{R}_{++}^{N\times 2}$, $m < 1 < M$ *and* $h = h_r : [m\,\bar{\delta}, M\,\bar{\delta}] \to \mathbb{R}_+$. *If* $m\,\bar{\delta} \preceq \delta \preceq M\,\bar{\delta}$ *with*

$$\delta = \bar{\delta} \cdot \left(\frac{\sum_{k=0}^{N-1} (b_k\, a_k^r)^{\frac{1}{(r+1)}}}{\sum_{k=0}^{N-1} b_k}\right)^{\frac{1}{r}} \cdot \left(b \odot a^{-1}\right)^{\frac{1}{r+1}}, \tag{27}$$

then δ *is optimal for* (17).

Remark 3.5: Corollary 3.5 simply tells that if δ from (27) is feasible for our master problem (17) under the choice $h = h_r$, then it must be optimal. For the sake of completeness, we also provide in Appendix A.4 closed-form schedules in what concerns an extended family of h_r functions. It includes the logarithmic model $h_0(\delta) = \log(\delta^{-1})$ from our introductory Example 1.1 as a pathological case $r \to 0$ and $N_\oplus, N_\ominus$ are not necessarily zero. As a drawback consequence, the employed notations become heavier.

Interpretation Rather intuitively, an iteration whose *impact coefficient* a_k is bigger harms more the upper-bound $\mathcal{E}(\{\delta_k\}_{k=0}^{N-1})$ on the objective gauge $\mathcal{C}(N)$ (cfr. (5)) thus requires more precision, i.e. a small δ_k. However its associated cost model $\mathcal{B}_k(\cdot) = b_k\, h(\cdot)$ (cfr. (6)) eventually reweights the accuracy of the oracle through b_k according to its relative computational burden with respect to the other iterations. Hence, δ_k^* are governed by *compound impacts* $\nu_k = b_k/a_k$,

$$\delta_k^* \propto \left(\frac{b_k}{a_k}\right)^{\frac{1}{r+1}}. \tag{28}$$

At fixed a, b, as r approaches 0, the oracles appear cheap and one can gain a lot by experiencing large deviations from $\bar{\delta}$. Conversely, we observe that if r tends to infinity then the cost of oracles spikes so that the constant schedule at $\bar{\delta}$ inexactness level, feasible, becomes optimal. Indeed, at any iteration, requesting an oracle accuracy even slightly better than $\bar{\delta}$ turns out to be extremely expensive and not affordable. Thus we must observe $\mathcal{T} = [N]$ if r is large enough. Another situation in which one can easily predict that $\mathcal{T} = [N]$ or, equivalently, $N_\ominus = 0 = N_\oplus$, arises when $m = 0$ and $M = \infty$. Indeed, akin to Example 3.4, the oracle cost blows up for $\delta \to m\bar{\delta} = 0$.

In addition, if not obliged to, one has no advantage to choose an inexactness parameter arbitrarily large $\delta \to M\bar{\delta} = \infty$.

Remark 3.6: Sticking to our main thread, let us express a possible asymptotical trend of optimal schedules of inexactness for Example 2.1 ($\mu = 0$, constant λ_k) based on (28)

$$\delta_k^* \propto A_{k+1}^{-\frac{1}{2}} \in \mathcal{O}(k^{-1})$$

Illustration We illustrate now the application of our previous theorems on an instance closely related to Example 2.1. Indeed, when $\mu = 0$ and $\lambda_k \in \mathcal{O}(1)$ for any integer $k \geq 0$, it is known that $a_k \in \Theta(k^2)$, see, e.g. [21]. We display the evolution of quantities $N_\oplus$ Figure 2 and δ_k^* Figure 3 with oracle's complexity parameter r and the factor of maximal tolerated inaccuracy M. Let $m = 0$ and let the *reference* inexactness tolerated be $\bar{\delta} = 10^{-4}$. In what follows, $N_\ominus = 0$ in any situation since the oracle cost model $h_r(\delta) = \delta^{-r}$ explodes as $\delta \to 0 = m\bar{\delta}$. Accordingly, we assume for any $k \geq 0$ that $b_k = 1$ and $a_k = (k+1)^2$. We highlight two observations of interest.

- Figure 2: $N_\oplus > 0$ if N itself is big enough and $M\bar{\delta}$ constrains our master problem (17), i.e. $M \to 1$. In this latter case, the computational savings on intend to invest in late iterations requiring more care must be spread out on more early iterations since the biggest δ_k from any optimal schedule $\{\delta_k^*\}_{k=0}^{N-1}$ cannot take a value that falls way above the reference $\bar{\delta}$. In other words, $M\bar{\delta}$ does not allow one to save a lot of efforts in the iterations linked with the smallest *impact factors*.
- Figure 3: The variability in the optimal schedules for inexactness $\{\delta_k^*\}_{k=0}^{N-1}$ heavily depends on the oracle cost parameter r, as emphasized by the power $(r+1)^{-1}$ in the

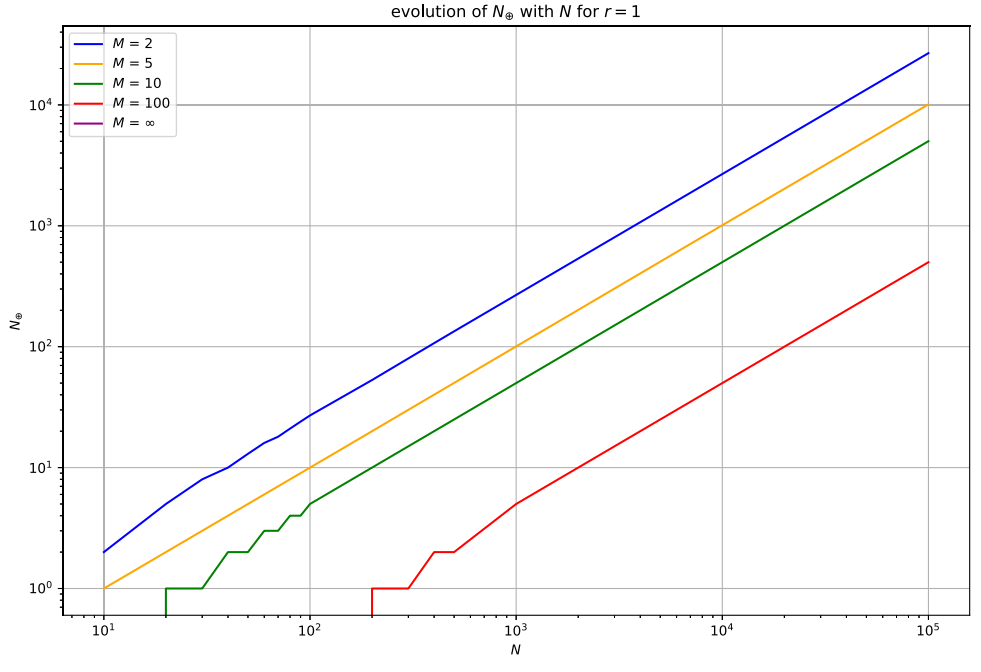


Figure 2. $N_\oplus$ grows with *selected* N, this effect starts earlier when (17) is more constrained, i.e. $M \to 1$.

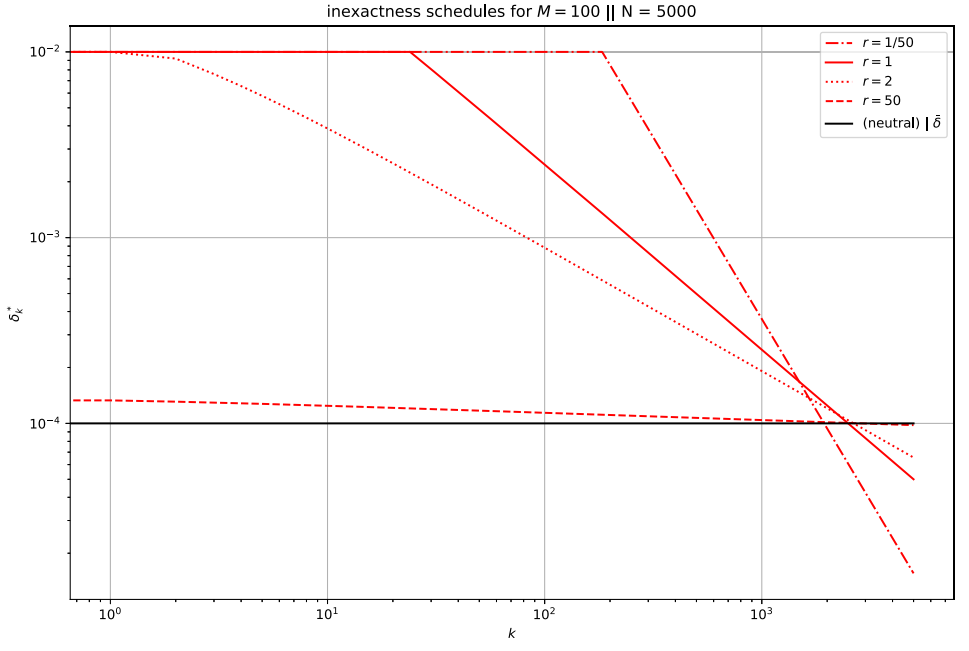


Figure 3. *N fixed,* $r \to 0$ means cheaper oracles and thus more aggressive schedules (see $r = 50^{-1}$).

relationship (28). Our findings discussed in the previous paragraph are graphically confirmed, e.g. when $r = 50$, $\delta_k^* \simeq \bar{\delta}$ for every $k \in \{0, \dots, N-1\}$.

3.3. Practical extensions

This part of our work is dedicated to direct extensions of the results unveiled so far.

We adapt them to practical scenarios beyond the initial scope of TOM. First, we investigate the modifications one should undertake to apply the concept of tunable oracle when, instead of the oracles' accuracies one would like to monitor the computational work invested in producing their outputs. Second and as previously hinted, we propose an *online* strategy which preserves the structure of optimal inexactness schedules without the knowledge of N.

3.3.1. Work controlled

In some cases, one would like to manually specify the time spent at each iteration. In other words, instead of deciding δ_k that would *a priori* lead to a cost of $\mathcal{B}_k(\delta_k)$, we process the other way around. We choose the amount of computations $\omega_k \in [\omega_M, \omega_m]$ and we expect to incur a level of inexactness $\delta_k \sim \mathcal{B}_k^{(-1)}(\omega_k)$. Therefore, we can equivalently fix a *reference* total work $\bar{\omega}$ that acts as a surrogate for the budget term $\sum_{k=0}^{N-1} \mathcal{B}_k(\bar{\delta})$. We rewrite the objective from master problem (17) as

$$\sum_{k=0}^{N-1} a_k \, \delta_k \sim \sum_{k=0}^{N-1} a_k \, \mathcal{B}_k^{(-1)}(\omega_k) = \sum_{k=0}^{N-1} a_k \, h_r^{(-1)}(\omega_k / b_k) \, .$$

Note the homogeneity of h_r allows to write for any $\eta > 0$, $\beta \geq 0$, $r > 0$ and $\Omega \subseteq \mathbb{R}_+^N$,

$$\arg\min_{\omega\in\Omega} \sum_{k=0}^{N-1} h_r^{(-1)}\left(\omega_k \cdot \eta / b_k\right) = \arg\min_{\omega\in\Omega} \sum_{k=0}^{N-1} h_r^{(-1)}\left(\omega_k / b_k\right). \tag{29}$$

This ensures that b_k must only be known up to a common constant multiplicative factor, as previously assumed. We formulate the work controlled counterpart of (17) in (30), assuming that $h = h_r$

$$\omega^*(N, \bar{\omega}, \omega_m, \omega_M) \in \arg\min_{\omega_M \preceq \omega \preceq \omega_m} \sum_{k=0}^{N-1} a_k\, h^{(-1)}\left(\frac{w_k}{b_k}\right) \quad \text{s.t.} \sum_{k=0}^{N-1} \omega_k = \bar{\omega}. \tag{30}$$

Let us state the analogous version of Corollary 3.5 regarding the work controlled framework. Again, the interested reader can look at Appendix A.4 that displays a full version with $N_\oplus$ and $N_\ominus$ not necessarily zero.

Corollary 3.6: *Let Assumptions* A *and* B *hold with* $(a, b) \in \mathbb{R}_{++}^{N\times 2}$, $\omega_M < \bar{\omega}/N < \omega_m$ *and* $h = h_r : \mathbb{R}_+ \to \mathbb{R}_+$. *If* $\omega_M \preceq \omega \preceq \omega_m$ *with*

$$\omega = \bar{\omega} \cdot \frac{(b \odot a^r)^{\frac{1}{r+1}}}{\sum_{k=0}^{N-1} (b_k\, a_k^r)^{\frac{1}{(r+1)}}}, \tag{31}$$

then ω *is optimal for* (30).

Remark 3.7: Despite being very similar, problems (17) and (30) are not perfectly equivalent in general. They do share the common goal of minimizing $\mathcal{E}$ subject to a budget constraint $\mathcal{B}$. Nevertheless, one can see (30) as a version of (17) where the bounds on the achievable δ_k at iteration $k \in \{0, \ldots, N-1\}$ vary. Indeed, we have the following bounds:

$$\delta_k \in \left[\mathcal{B}_k^{(-1)}(\omega_m),\, \mathcal{B}_k^{(-1)}(\omega_M)\right]. \tag{32}$$

However, in the neutral case where $b \propto \mathbf{e}$, these bounds are constant and can be written as $[m\bar{\delta},\, M\bar{\delta}]$ for any $\bar{\delta} > 0$ and well-chosen $0 < m \leq M$ parameters.

Interpretation Just as in (28), it is not difficult to construe results from Corollary 3.6 by advocating that

$$\omega_k^* \propto \left(b_k\, a_k^r\right)^{\frac{1}{r+1}}. \tag{33}$$

We have already seen that when a_k grows, linked oracle's accuracy δ_k should evolve inversely proportional. As a consequence, the computational cost ω_k rise accordingly. One should pay attention to the role of b_k in (33). It seems like a bigger b_k implies a bigger ω_k, which turns out to be true. Meanwhile, a bigger b_k is prone to curb the demand for highly accurate oracles in (28) hence intuitively reducing oracle's work ω_k. Yet, there is no contradiction. Combining (28) and Assumption B, we can capture the overall marginal effect of b_k in

$$\omega_k = b_k h(\delta_k) \propto b_k \left((b_k/a_k)^{\frac{1}{r+1}}\right)^{-r} = b_k^{\frac{1}{r+1}} a_k^{\frac{r}{r+1}}.$$

Finally, let us assert that b_k's *weighting impact* diminishes as $r \to \infty$, ω_k^* becoming asymptotically proportional to the *impact coefficient* a_k. Conversely when $r \to 0$, the suggested

oracle cost somehow flattens. It ultimately depends only on b_k, constant in the absence of further knowledge (see Examples 2.1 and 2.3).

3.3.2. Online version

So far, we have assumed that the choice of N was exogenous and well thought. Unfortunately, there is no *one fits all* approach to adequately fix N. Usually, one runs an optimization algorithm and stops it as soon as a relative tolerance, tracked alongside the iterations, is observed. Let δ^* (respectively ω^*) be an optimal solution of (17) (respectively (30) for a specified $N \in \mathbb{N}$ and $h = h_r$. Hidden behind equation (27) (respectively (31)) and emphasized by (28) (respectively (33)), one can retrieve a recursion rule linking the optimal inexactness parameters of two distinctive iterations, say $\hat{k}$ and k. Let $\delta_k^* \in [m\bar{\delta}, M\bar{\delta}]$ then for any $\hat{k} \in [N]$, one can recover

$$\delta_{\hat{k}}^* = \max\left\{m\bar{\delta}, \min\left\{M\bar{\delta}, \left(\frac{b_{\hat{k}}}{a_{\hat{k}}} \cdot \frac{a_k}{b_k}\right)^{\frac{1}{(r+1)}} \delta_k^*\right\}\right\}. \tag{34}$$

Following the same logic, a recursion exists for $\omega_k^* \in [\omega_M, \omega_m]$,

$$\omega_{\hat{k}}^* = \max\left\{\omega_M, \min\left\{\omega_m, \left(\frac{b_{\hat{k}}\, a_{\hat{k}}^r}{b_k\, a_k^r}\right)^{\frac{1}{(r+1)}} \omega_k^*\right\}\right\}. \tag{35}$$

Therefore, in practice, one can adopt a *reference* situation accounting for a lower bound of N_r iterations then compute its inherent *offline* optimal schedule. For any iteration $\hat{k} \geq N_r$, one can extrapolate using the recursion rules explained right above.

Rationale. The extrapolated schedules obtained with (34), (35) present the advantage to preserve the right ratios $\delta_{\hat{k}}^*/\delta_k^*$ and $\omega_{\hat{k}}^*/\omega_k^*$ as if we knew the number of iterations performed by our Tunable Oracles Method (TOM) when exited. In addition, it does not break our practical assumption of relative knowledge of a_k's and b_k's since we only involve ratios wiping out any common multiplicative constant. Finally, such *online* schedule allows for b_k/a_k's that are not necessarily pre-computed. As mentioned earlier, in various adaptive methods such coefficients are defined by the final result of local line-search techniques [5,13,21]. This feature of our *online* strategy dramatically extends the applicability of our approach, endowing it with local information exploitation, allowing for possible *non-monotonicity* in the parameters δ_k or ω_k used.

4. Numerical experiments

We present three experiments that showcase both our elaborated *offline* and *online* techniques for Tunable Oracle Methods (TOM). We emphasize that we are primarily concerned with showing that our approach effectively improves the efficiency of known existing methods such as FGM [21].

The first experiment serves to validate our theoretical optimal *offline* schedules for various levels of parameters, i.e. r (oracle cost parameter), $\bar{\delta}$ (*reference* inexactness) and N

(number of performed iterations). To stick to our theoretical framework as much as possible, we generate inexact oracle outputs by adding artificial noise to the gradients used in FGM. The noise is chosen according to a simulated oracle cost one would invest to control the level of inexactness. Confirming that our optimal schedules also perform better in a practical setting is important, as numerical optimization methods (with or without inexactness) typically perform better than their worst-case guarantees in practice. Within this setting, we show the superiority of our schedules compared to a *constant* schedule approach with matching overall computational cost.

In the second and third experiments, we investigate how the *offline* and our heuristic *online* approach behave within a real case where inexactness naturally emerges. Experiments 1, 2 and 3 rely on the problem motivated in Example 2.3 but Experiment 3 allows for line search within FGM hence turning it into an *online* method.

Our code is freely available on GitHub so that one can observe that the reported results are representative of the usual performances.

Test problem and data generation

We describe a *robust optimization* task involving a regularization parameter $\mu \geq 0$ and data vectors $\Theta = \{\theta_i\}_{i=1}^n$,

$$\theta_i \sim \mathcal{N}\left(\mathbf{0}, I_d/p\right) \quad \forall\, i \in \{1, \ldots, n\}$$

a collection of $n \in \mathbb{N}$ *scenarios* from $\mathbb{R}^d$ ($p > 0$). We consider the classical problem where one minimizes the worst outcome of the regularized linear objective only over the previously seen *scenarios*, i.e. one wishes to minimize a regularized objective $\frac{\mu}{2}||x||^2 + \langle \theta, x\rangle$ over the unit simplex, i.e. $\Delta := \{x \in \mathbb{R}^d \,|\, x \succeq \mathbf{0},\ ||x||_1 = 1\}$ for a meaningful cost θ based on historical data about its value, stored in $O = [\theta_1, \ldots, \theta_n]^T$,

$$\min_{x\in\Delta} \max_{\theta\in\Theta} \langle \theta, x\rangle + \frac{\mu}{2}||x||^2 = \min_{x\in\Delta} \max_{\theta\in\mathbf{conv}(\Theta)} \langle \theta, x\rangle + \frac{\mu}{2}||x||^2. \tag{36}$$

4.1. Experiment 1 — softmax optimization under synthetic noise

In a large-scale setting, solving (36) using the standard epigraph reformulation might be prohibitive and is sometimes replaced by a smoothed version of the robust objective, leaving the feasible set intact, i.e. no extra constraint. Let $\upsilon > 0$, the problem becomes[2]

$$F^* = \min_{x\in\mathbb{R}^d} \overbrace{\underbrace{\upsilon^{-1} \log\left(n^{-1} \sum_{i=1}^{n} \exp(\upsilon \cdot \langle \theta_i, x\rangle)\right) + \frac{\mu}{2}||x||^2}_{f(x)} + \underbrace{\chi_\Delta(x)}_{\Psi(x)}}^{F(x)}. \tag{37}$$

Solving (37) using FGM requires the ability to compute (in-)exact information about f at any query point $x \in X := \Delta$, i.e. $(\tilde{f}(x), \nabla\tilde{f}(x)) \simeq (f(x), \nabla f(x))$. Here, the value of $f(x)$ and its gradient $\nabla f(x)$ are available in closed form for (37) hence discarding the need for schedules of inexactness. However, one could think about a game where one has to pay

Table 1. Illustration of Assumptions A and B.

	$h(\delta_k)$	$\propto a_k$	$\propto b_k$
Example 2.1	δ_k^{-1}	$A_{k+1}\cdot(1+\mu\lambda_k)^2\cdot\lambda_k^{-1}$	1
Example 2.2	$\delta_k^{-\frac{2}{3}}$	t_k^{-1}	$t_k^{\frac{2}{3}}$
Example 2.3	$\log(\delta_k^{-1})/\delta_k^{-\frac{1}{2}}$	A_{k+1}	1

δ^{-r} $(r>0)$ or $-\log(\delta)$ $(r=0)$ to obtain $\nabla f(x)$ corrupted by a noise of limited radius $\alpha\,\delta$ depending on parameter $r\geq 0$, i.e. one gets $\nabla\tilde{f}(x)$ such that

$$\nabla\tilde{f}(x)=\nabla f(x)+\alpha\,\delta\cdot u, \tag{38}$$

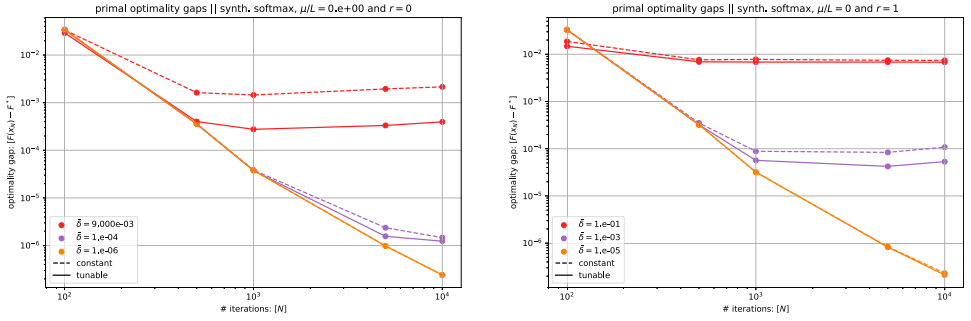
with $u\sim\mathcal{U}(\{u\in\mathbb{R}^d\,|\,||u||=1\})$. From [3], one can deduce that $(f(x),\nabla\tilde{f}(x))$ as defined above provides a $(4\,\alpha\,\delta,\upsilon\cdot||O||^2+\mu,\mu)$ inexact information.

Test instances As suggested by Example 2.3, let us now try out FGM on problem (37) with fixed stepsize $L^{-1}=(\upsilon\cdot||O||^2+\mu)^{-1}$ and different values of μ, r, N, $\bar{\delta}$ while using sequences of inexact information tuples parametrized by schedules $\{\delta_k\}_{k=0}^{N-1}$. That is, for each instance $(\mu,r,N,\bar{\delta})\in\{0,10^{-1}\}\times\{1\}\times\{10,500,1000,5000,10000\}\times\{10^{-1},10^{-3},10^{-5}\}$ or $(\mu,r,N,\bar{\delta})\in\{0,10^{-1}\}\times\{0\}\times\{10,500,1000,5000,10000\}\times\{9\cdot 10^{-3},10^{-4},10^{-6}\}$, we compare our *offline* optimized approach $\{\delta_k=\delta_k^*\}_{k=0}^{N-1}$ where δ^* solves (17) with vectors (a,b) specified in Table 1, $m=0$ and $M=100$ (we allow the inexactness from our *tunable* approach to be a hundred times worse than the *reference* $\bar{\delta}$) with the *constant* schedule $\{\delta_k=\bar{\delta}\}_{k=0}^{N-1}$. We define the simulated cost of inexact oracles as $N/\bar{\delta}=\sum_{k=0}^{N-1}(\delta_k^*)^{-1}$ $(r=1)$ and $-N\log(\bar{\delta})=-\sum_{k=0}^{N-1}\log(\delta_k^*)$ $(r=0)$[3] and match it to the total cost of the constant $\bar{\delta}$ *reference* schedule. Finally, to tame the effects of the randomness inherent to our synthetic inexact information production, we aggregate the obtained results from FGM over 5 runs with 5 different starting points drawn as $x_0\sim\mathcal{U}(\Delta)$. Here below, the dimensions read $(d,n,\alpha,p)=(100,500,100,10)$.

Estimation of F^* Regarding this simple example, F^* is estimated by the value of F at the output from a noise-free FGM applied for $3\cdot 10^5$ iterations.

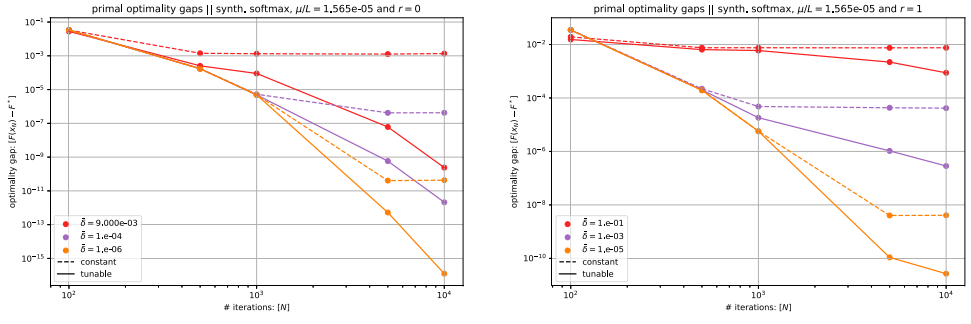
Results Figures 4 and 5 show the evolution of the terminal primal optimality gap $F(x_N)-F^*=f(x_N)-F^*$ after N iterations of FGM (output iterate x_N is in Δ) using the variants of inexactness schedules described above. Each dot plotted on these graphs represents the averaged performance over the $25=5\cdot 5$ independent runs of either our *tunable* or the *constant* approach for a single instance $(\mu,r,N,\bar{\delta})$.

As predicted by the theory, the *tunable* inexactness schedule reached a better terminal primal optimality gap for every setting tried. Not surprisingly, the gains were dramatically more impressive when the discrepancy within *impact coefficients* $\{a_k\}_{k=0}^{N-1}$ was high. Intuitively, when such coefficients do not vary at all, our optimized *tunable* approach boils down to the *constant* schedule and no gain is to expect at all. Table 1 recalls that $a_k=A_{k+1}$ for any $k\in[N]$ and Remark 3.2 suggests that $\{A_k\}_{k\in\mathbb{N}}$ grows sublinearly (respectively linearly) when $\mu=0$ (respectively $\mu>0$). Therefore, settings in which $\mu>0$ are the most promising in terms of gains in favour of the *tunable* approach, thanks to the inherent high variance of *impact coefficients*. Figure 5 undoubtedly confirmed this hope with substantial gains for our *tunable* approach although problem's conditioning stays relatively small, i.e.



(a) logarithmic oracle cost ($r = 0$) (b) inversely proportional oracle cost ($r = 1$)

Figure 4. (No regularization, $\mu = 0$) — Softmax optimization with synthetic noise: (a) logarithmic oracle cost ($r = 0$) and (b) inversely proportional oracle cost ($r = 1$).



(a) logarithmic oracle cost ($r = 0$) (b) inversely proportional oracle cost ($r = 1$)

Figure 5. (Regularization, $\mu > 0$) — Softmax optimization with synthetic noise: (a) logarithmic oracle cost ($r = 0$) and (b) inversely proportional oracle cost ($r = 1$).

$\mu/L \simeq 10^{-5}$. Both Figures 4 and 5 show that the benefits of high discrepancy in *impact coefficients* (hence in optimal inexactness schedules, see (28)) are enhanced by cheap oracles, i.e. $r \to 0$. We point out as a generic comment that for a small number of iterations, the additive effect of inexactness does not dominate the *error-free* convergence bound and one can barely observe any difference between our optimized approach and other inexactness schedules. Finally, one can observe that some of the results linked to more accurate *reference* inexactness of $\bar{\delta} \leq 10^{-5}$ led to poorer primal optimality gaps for $N = 10^2$. This unexpected phenomenon can be explained by the fact that for bigger values of δ, (38) might actually yield more aggressive and successful descent directions, compared to plain accurate gradients, again in a regime where the additive impact of inexactness is of secondary importance.

4.2. Experiment 2 & 3 — robust optimization over convex hull

As depicted in Example 2.3, one may choose an *anchor scenario* $\bar{\theta} \in \mathbf{conv}(\Theta)$ and a parameter $\sigma > 0$ to regularize the inner maximization problem in (36) (right-hand side) yielding a smoothed outer level objective [16]. Then, a challenging task consists of minimizing the

worst possible outcome of the inner-regularized objective over the convex hull of all the *scenarios*, i.e.

$$\min_{x\in\Delta}\max_{\theta\in\mathbf{conv}(\Theta)}\langle\theta,x\rangle-\frac{\sigma}{2}||\theta-\bar{\theta}||^2+\frac{\mu}{2}||x||^2. \tag{39}$$

To that end, we adopt the *anchor* $\bar{\theta}=n^{-1}\sum_{i=1}^{n}\theta_i$ and construct the problem

$$F^*=\min_{x\in\mathbb{R}^d}\overbrace{\underbrace{\frac{\mu}{2}||x||^2+\max_{w\succeq\mathbf{0},||w||_1=1}\underbrace{\langle O^Tw,x\rangle-\frac{\sigma}{2}||O^Tw-\bar{\theta}||^2}_{q(w;x)}}_{f(x)}+\underbrace{\chi_\Delta(x)}_{\Psi(x)}}^{F(x)}. \tag{40}$$

Again, as motivated in Example 2.3, we use FISTA from [23] with a momentum depending on the number $\hat{\kappa}=\frac{\lambda_{\min}(OO^T)}{\lambda_{\max}(OO^T)}$ in order to get $w_x\succeq\mathbf{0}$, $||w_x||_1=1$ ensuring

$$f(x)-\left(\frac{\mu}{2}||x||^2+q(w_x;x)\right)\leq\delta. \tag{41}$$

Then $(\tilde{f}(x),\nabla\tilde{f}(x))=(\frac{\mu}{2}||x||^2+q(w_x;x),\mu x+O^Tw_x)$ provides $(2\delta,2\cdot\sigma^{-1}+\mu,\mu)$ inexact information about f, tuple used within FGM to solve problem (39).

We monitor the quality of a candidate $w_x^{(\omega)}$ at any point $x\in\Delta$ after ω inner iterations thanks to the Frank–Wolfe gap $\mathrm{FW}(\omega)$, which is defined as follows:

$$\mathrm{FW}(\omega)=\max_{j\in[\omega]}\max_{w\succeq\mathbf{0},||w||_1=1}q(w_x^{(j)};x)+\langle\nabla q(w_x^{(j)};x),w-w_x^{(j)}\rangle-q(w_x^{(\omega)};x). \tag{42}$$

As soon as the Frank–Wolfe gap satisfies $\mathrm{FW}(\omega)\leq\delta$ we have that the quality $w_x^{(\omega)}$ fulfils (41). In order to speed-up computations to obtain approximate solutions of the inner problem, we give as starting point of FISTA the solution obtained at the last oracle call.

Estimation of F^* Let $\hat{x}^*\in\Delta$ be the best known solution to (40). Then, by convexity of the objective, we estimate a lower-bound for F^* as follows:

$$F^*=\min_{x\in\Delta}f(x)\geq\min_{x\in\Delta}\underbrace{\frac{\mu}{2}||\hat{x}^*||^2+q(w_{\hat{x}^*};\hat{x}^*)}_{\tilde{f}(\hat{x}^*)}+\underbrace{\langle\mu\hat{x}^*+O^Tw_{\hat{x}^*}}_{\nabla\tilde{f}(\hat{x}^*)},x-\hat{x}^*\rangle+\frac{\mu}{2}||x-\hat{x}^*||^2, \tag{43}$$

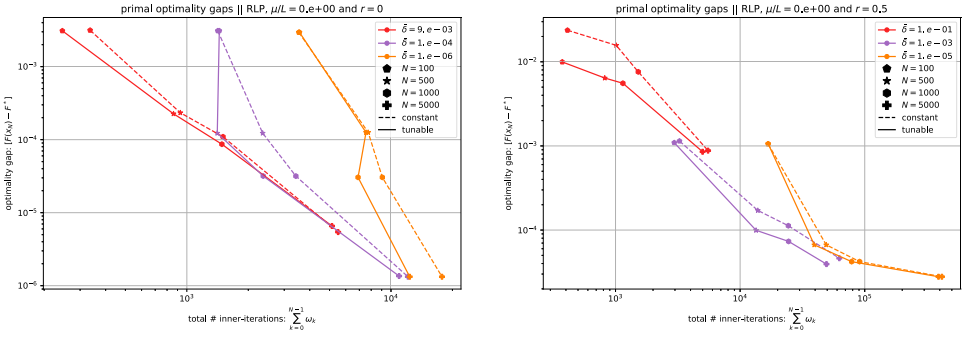
where $w_{\hat{x}^*}$ represents the w candidate that achieved 10^{-10} precision when computing an approximate pair $(\tilde{f}(\hat{x}^*),\nabla\tilde{f}(\hat{x}^*))\simeq(f(\hat{x}^*),\nabla f(\hat{x}^*))$ with respect to criterion (41).

Offline schedules (Experiment 2)

Test instances Here, we run FGM with stepsize $L^{-1}=(2\sigma^{-1}+\mu)^{-1}$ for a predefined number of iterations N. We have tested the two following settings in terms of scaling and dimension: $(d,n,p,\sigma)=(100,500,5^{-1},10^{-3})$ and $(d,n,p,\sigma)=(1000,500,5^{-1},10^{-3})$. The first setting led to $\hat{\kappa}=0$ hence implying that the oracle cost parameter r equals $\frac{1}{2}$ whereas under the second setting $\hat{\kappa}>0$ and r is taken as 0, recalling the arguments of Example 2.3. For each instance $(\mu,r,N,\bar{\delta})\in$

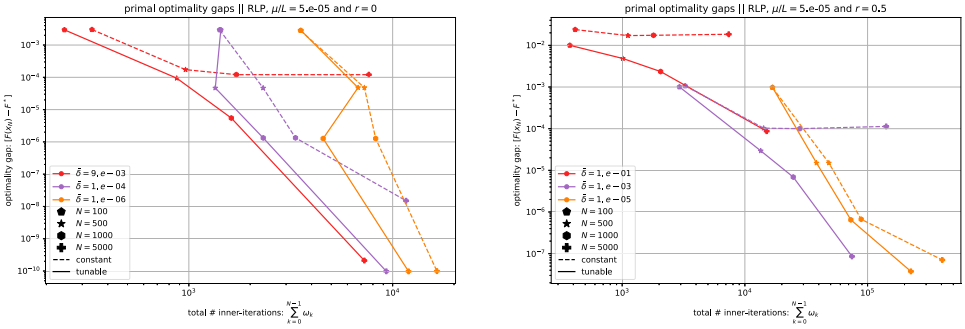
$\{0, 10^{-1}\} \times \{\frac{1}{2}\} \times \{10, 500, 1000, 5000\} \times \{10^{-1}, 10^{-3}, 10^{-5}\}$ or $(\mu, r, N, \bar{\delta}) \in \{0, 10^{-1}\} \times \{0\} \times \{10, 500, 1000, 5000\} \times \{9 \cdot 10^{-3}, 10^{-4}, 10^{-6}\}$, we obtain our *tunable* δ^* by solving problem (17) in the very same fashion as Experiment 1. Although the overall computational cost of our *tunable* $\{\delta_k = \delta_k^*\}_{k=0}^{N-1}$ schedule is meant to coincide with the computational cost of the *constant* schedule $\{\delta_k = \bar{\delta}\}_{k=0}^{N-1}$, the oracle cost model does not always perfectly fit the true encountered computational costs required to produce the inexact information tuples. Therefore, we record the total number of inner iterations $\sum_{k=0}^{N-1} \omega_k$ as well as the final primal accuracy about (40) for fair comparisons. This latter entails the computation of $f(x_N) = F(x_N)$, also performed at 10^{-10} precision with respect to criterion (41) (see above). Here, $\hat{x}^*$ stems as the output of FGM for $8 \cdot 10^3$ iterations with *constant* inexactness level of 10^{-8} when $r = 0$. Because of much more expensive oracles, we execute only $5 \cdot 10^3$ iterations at inexactness level 10^{-7} to produce $\hat{x}^*$ when $r = \frac{1}{2}$. To reduce the eventual initialization bias $x_0 \sim \mathcal{U}(\Delta)$, we averaged the results of 10 repetitions.

Results Here, we have plotted the terminal primal optimality gaps against the overall oracle workloads $\sum_{k=0}^{N-1} \omega_k$ after N (outer-)iterations of FGM. Since the wall-clock times strongly correlate with the workloads, we have chosen no to display duplicate figures other than Figures 6 and 7. Again, each dot symbolizes the averaged performance tracked for one specific inexactness schedule for one specific instance defined by the vector $(\mu, r, N, \bar{\delta})$. One can observe that most of the trends highlighted in the paragraph about the results of our synthetic first experiment still hold for this real second experiment. Notably, it looks like that regularization plays a key role in the substantial superiority of our optimized *tunable* approach. Again, this is understood by a variability of possibly several orders of magnitude in the value of *impact coefficients* $\{a_k\}_{k=0}^{N-1}$ when $\mu > 0$. In every case, our *tunable* approach was more efficient, i.e. it dominated the *constant* approach in the Pareto sense in the plane of reached primal optimality versus unit of workload spent in the inner problems. In Figures 6(a) and 7(a), the *tunable* schedule based on a *reference* inexactness $\bar{\delta}$ fixed to 10^{-6} (yellow) did take more inner iterations for $N = 500$ than $N = 1000$. Although not intuitive, this behaviour can be explained. It happens that oracles cost way less than expected, especially for low requested accuracies. In what concerns Experiment 2, it could



(a) smooth strongly convex inner problem ($r = 0$) (b) smooth convex inner problem ($r = \frac{1}{2}$)

Figure 6. (No regularization, $\mu = 0$) — R.O. over Convex Hull, *offline* with multiple $\bar{\delta}$: (a) smooth strongly convex inner problem ($r = 0$) and (b) smooth convex inner problem ($r = \frac{1}{2}$).



(a) smooth strongly convex inner problem ($r = 0$) (b) smooth convex inner problem ($r = \frac{1}{2}$)

Figure 7. (Regularization, $\mu > 0$) — R.O. over Convex Hull, *offline* with multiple $\bar{\delta}$: (a) smooth strongly convex inner problem ($r = 0$) and (b) smooth convex inner problem ($r = \frac{1}{2}$).

even happen that FISTA's starting iterate at outer iteration k, $w_{x_k}^{(0)} = w_{x_{k-1}}^{(\omega_{k-1})}$, already fulfils δ_k inexactness for free, i.e. $\omega_k = 0$. It turns out that our *tunable* approach for $N = 1000$ benefited more of such *cheap-meal* phenomenons since its linked schedule demands less accurate oracles than the one with $N = 500$ in early iterations. Overall, a *reference* inexactness $\bar{\delta} \in [10^{-3}, 10^{-2}]$ provided the best *trade-off* to reach a given primal accuracy target within this *offline* setting.

Online schedules (Experiment 3)

Test instances Finally, we run FGM with adaptive stepsizes $L_k^{-1} > 0$ for any $k \geq 0$. A stepsize is validated at iteration k as soon as the inequality

$$\tilde{f}(x_{k+1}) \leq \tilde{f}(y_k) + \langle \nabla \tilde{f}(y_k), x_{k+1} - y_k \rangle + \frac{L_{k+1}}{2} ||x_{k+1} - y_k||^2 + 2\,\delta_k \tag{44}$$

holds for $(2\delta_k, \sigma^{-1} + \mu, \mu)$ inexact information tuples at feasible points x_{k+1} and y_k respectively. FGM increases by 3/2 the stepsize when an iteration succeeds, i.e. (44) passes, and decreases it by a factor 2 otherwise. Note that (44) is always satisfied for $L_{k+1} \geq \sigma^{-1} + \mu$. Convergence guarantees like (13) are conserved [21] but now the *impact coefficients* $a_k = A_{k+1}$ depend on L_{k+1} via the recursion ($A_0 = 0$)

$$A_{k+1}(1 + \mu A_k) = L_{k+1}(A_{k+1} - A_k)^2 \tag{45}$$

and are therefore not predictable in advance. This paves the way for the application of our *online* approach (34) with $N_r = 50$. N_r is kept small in practice, its sole interest being to specify the very first inexactness parameters before the *online* approach takes over. Yet, one must simulate the N_r first coefficients $\{a_k\}_{k=0}^{N_r-1}$ required to devise the *offline* optimized schedule $\{\delta_k^*\}_{k=0}^{N_r-1}$ from (17). To that purpose, one fakes the value of L_{k+1} fixing it to $L = \sigma^{-1} + \mu$ within (45) and uses the computed $\{a_k = A_{k+1}\}_{k=0}^{N_r-1}$.

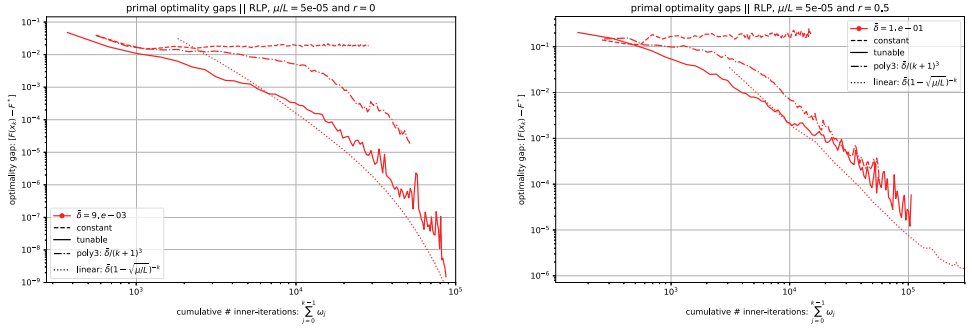
For each level of *reference* inexactness $\bar{\delta} \in \{9 \cdot 10^{-3}, 10^{-4}, 10^{-6}\}$, we tried four different inexactness schedules. That is for any iteration index $k \geq 0$, the *constant* one $\delta_k = \bar{\delta}$, the heuristic *online tunable* (as described above) with $a_k = A_{k+1}$, the fully sublinear monotonically decreasing tuned as in [24] (`poly-3`), $\delta_k = \bar{\delta}\,(k+1)^{-3}$ and the fully linear one

[13] (`linear`) with $\delta_k = \bar{\delta}\,(1 - \sqrt{\mu/L})^{-k}$. For this experiment, $\hat{x}^*$ (see (43)) is taken as the output of adaptive FGM after at $3 \cdot 10^3$ iterations with *constant* inexactness of 10^{-12} for $r = 0$ and 10^{-8} otherwise for $r = \frac{1}{2}$. We have conducted five random initializations of the settings; $(d, n, p, \sigma) = (50, 100, 5^{-1}, 10^{-3})$ which, akin to Experiment 2, led to $\hat{\kappa} = 0$ and thus $r = \frac{1}{2}$ and $(d, n, p, \sigma) = (200, 100, 5^{-1}, 10^{-3})$ that gave $\hat{\kappa} > 0$ and, accordingly, $r = 0$. Finally, we set the regularization to $\mu = 10^{-1}$.

Results We decided to keep track of the primal objective value every 10 outer iterations, i.e. when $k \bmod 10 = 0$, we did compute $\tilde{f}(x_k) \simeq f(x_k)$ with precision 10^{-10}. After the runs, we turned these primal objective values into primal optimality gaps thanks to our estimation of F^*.

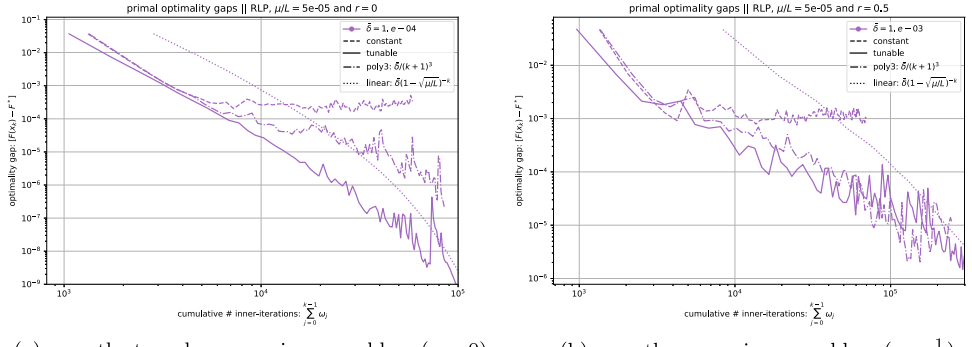
We can observe on Figures 8, 9 and 10 that our *online* heuristic *tunable* inexactness schedule adapts well to the *error-free* speed of convergence of FGM, symbolized by the value of the main sequence $\{A_k\}_{k\in\mathbb{N}}$.

We recall that its starting growth is of order $\mathcal{O}(k^2)$ but eventually, since $\mu > 0$ in the present experiment, a linear growth rate shows up lately [13,21]. It is worth noting that



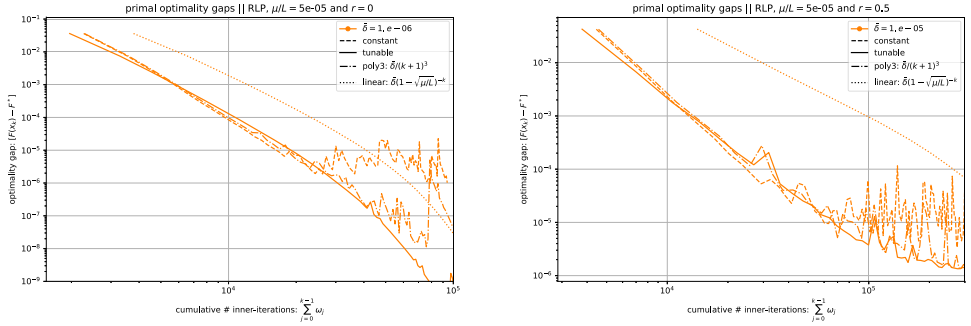
(a) smooth strongly convex inner problem ($r = 0$) (b) smooth convex inner problem ($r = \frac{1}{2}$)

Figure 8. (Regularization, $\mu > 0$) — R.O. over Convex Hull, *online* with low accuracy $\bar{\delta}$: (a) smooth strongly convex inner problem ($r = 0$) and (b) smooth convex inner problem ($r = \frac{1}{2}$).



(a) smooth strongly convex inner problem ($r = 0$) (b) smooth convex inner problem ($r = \frac{1}{2}$)

Figure 9. (regularization, $\mu > 0$) — R.O. over Convex Hull, *online* with medium accuracy $\bar{\delta}$: (a) smooth strongly convex inner problem ($r = 0$) and (b) smooth convex inner problem ($r = \frac{1}{2}$).



(a) smooth strongly-convex inner problem ($r = 0$) (b) smooth convex inner problem ($r = \frac{1}{2}$)

Figure 10. (Regularization, $\mu > 0$) — R.O. over Convex Hull, *online* with high accuracy $\bar{\delta}$: (a) smooth strongly-convex inner problem ($r = 0$) and (b) smooth convex inner problem ($r = \frac{1}{2}$).

(`linear`) schedule was the best in the low *reference* inexactness regime. Nevertheless, overall, neither (`poly-3`) or (`linear`) adapts as well as our schedule. Indeed, although comparable to *tunable* when $r = \frac{1}{2}$ on Figures 8(b), 9(b) and 10(b), `poly-3` does not take advantage of the regularization whereas `linear` appears way too conservative on Figures 9 and 10, perhaps to preserve as much as possible the asymptotical rate of convergence of the *error-free* counterpart of FGM. The reference *constant* schedule for inexactness turns out to be competitive until *error-accumulation* (see Remark 2.1) undermines further improvements.

5. Conclusion

In this paper, we considered a class of iterative algorithms, namely Tunable Oracle Methods (TOM), for which one can take advantage from the combined knowledge of both the computational cost associated with the oracle calls and the impact of associated inexactness on the convergence. We have shown how to choose optimally the level of inexactness that should requested at each iteration. Our numerical experiments confirm the superiority of these optimal schedules over the use of constant inexactness, for a given total computational budget, and also compare favourably with existing baselines from the literature. Future work may include a *tight analysis* of other iterative methods that involve controllable inexactness, e.g. bilevel learning, for which we can also hope that optimal inexactness schedules enhance the practical performances. Another direction for future research would be to gain insight about *random inexactness* rather than the (possibly) adversarial model considered in this paper. In this context, the goal could be to choose the appropriate amount of work to drive the distribution of inexactness towards the local needs of the iterative algorithm.

Notes

1. The existence of a closed-form or at the expense of an easy one-dimensional segment-search.
2. Note that the value of (37) differs at most by $\epsilon = \mu + n/\upsilon$ from (36).
3. $M\,\bar{\delta} < 1$ to ensure that any iteration costs a strictly positive amount of work.

Acknowledgments

The authors are grateful to Yurii Nesterov and Pierre-Antoine Absil for their advice both in terms of content and presentation of the results.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work was supported in part by the Fonds de la Recherche Scientifique - FNRS and the Fonds Wetenschappelijk Onderzoek - Vlaanderen under EOS Project 30468160. Guillaume Van Dessel was funded by UCLouvain as a teaching assistant.

Notes on contributors

Guillaume Van Dessel earned his bachelor degree in electrical and mathematical engineering in 2017. Then, he graduated in Data Sciences Engineering from UCLouvain in 2019. During his master, he had the opportunity to spent five months at EPFL in academic exchange where he joined the Machine Learning and Optimization (MLO) research group. He enrolled as a teaching assistant in his alma mater UCLouvain, occupying a full-time job in the applied mathematics department. His main research topics are devoted to numerical methods dedicated to optimization, with quite an emphasis on structured nonconvex problems and inexact black-box settings. He is also interested in mathematical models and numerical simulations.

François Glineur earned dual engineering degrees from CentraleSupélec and Université de Mons in 1997, and a PhD in Applied Sciences from the latter institution in 2001. After visiting Delft University of Technology and working as a post-doctoral researcher at McMaster University, he joined Université catholique de Louvain, where he is currently a full professor of applied mathematics in the Engineering School. He is also a member of the Center for Operations Research and Econometrics and the Institute of Information and Communication Technologies, Electronics and Applied Mathematics. He served as the Engineering School's vice-dean between 2016 and 2020. His primary areas of interest in research are optimization models and methods, with a particular emphasis on convex optimization and algorithmic efficiency. He received the 2017 Optimization Letters best paper award for the worst-case analysis of gradient descent with line search. He is also interested in the use of optimization in engineering, as well as nonnegative matrix factorization and its application to data analysis.

References

[1] A. Chambolle and T. Pock, *An introduction to continuous optimization for imaging*, in *Cambridge University Press (CUP)*, Vol. 25, Acta Numerica, 2016, pp. 161–319.
[2] O. Devolder, *Exactness, inexactness and stochasticity in first-order methods for large-scale convex optimization*, in *Doctoral Thesis*, 2013.
[3] O. Devolder, F. Glineur, and Y. Nesterov, *First-order methods with inexact oracle: the strongly convex case*. LIDAM Discussion Papers CORE 2013016, Université catholique de Louvain, Center for Operations Research and Econometrics (CORE), 2013.
[4] N. Doikov and Y. Nesterov, *Inexact tensor methods with dynamic accuracies*, in *ICML*, 2020.
[5] D. Drusvyatskiy and C. Paquette, *Efficiency of minimizing compositions of convex functions and smooth maps*, Math. Program. 178 (2019), pp. 503–558.
[6] N. Hallak and M. Teboulle, *Finding second-order stationary points in constrained minimization: A feasible direction approach*, J. Optim. Theory. Appl. 186 (2020), pp. 480–503.

[7] A. Kavis, K.Y. Levy, F.R. Bach, and V. Cevher, *Unixgrad: A universal, adaptive algorithm with optimal guarantees for constrained optimization*, in *NeurIPS*, 2019.
[8] M.S. Kodialam and H. Luss, *Algorithms for separable nonlinear resource allocation problems*, Oper. Res. 46(2) (1998), pp. 272–284.
[9] J.M. Kohler and A. Lucchi, *Sub-sampled cubic regularization for non-convex optimization*, in *ICML*, 2017.
[10] W. Kong and R.D.C. Monteiro, *An accelerated inexact dampened augmented Lagrangian method for linearly-constrained nonconvex composite optimization problems*, Comput. Optim. Appl. 85 (2023), pp. 509–545.
[11] J. Lu and M. Johansson, *Convergence analysis of approximate primal solutions in dual first-order methods*, SIAM J. Optim. 26(4) (2016), pp. 2430–2467.
[12] P. Machart, S. Anthoine, and L. Baldassarre, *Optimal computational trade-off of inexact proximal methods. [Research Report] Aix-Marseille Université*, hal-00704398v3, 2012.
[13] M. Barré, A. Taylor, and F. Bach, *A note on approximate accelerated forward-backward methods with absolute and relative errors, and possibly strongly convex objectives*, Open J. Math. Optim. 3 (2022), pp. 1–15.
[14] A. Mokhtari, A. Ozdaglar, and A. Jadbabaie, *Escaping saddle points in constrained optimization*, in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, Red Hook, NY, USA, Curran Associates Inc., 2018, pp. 3633–3643.
[15] R.D.C. Monteiro and B.F. Svaiter, *An accelerated hybrid proximal extragradient method for convex optimization and its implications to second-order methods*, SIAM J. Optim. 23(2) (2013), pp. 1092–1125.
[16] Y. Nesterov, *Smooth minimization of non-smooth functions*, Math. Program. 103 (2005), pp. 127–152.
[17] Y. Nesterov, *Gradient methods for minimizing composite functions*, Math. Program. 140(1, Ser. B) (2013), pp. 125–161.
[18] B. Pötzl, A. Schiela, and P. Jaap, *Inexact proximal newton methods in Hilbert spaces*. Comput. Optim. Appl. 87 (2024), pp. 1–37.
[19] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach, *Proximal methods for sparse hierarchical dictionary learning*, in *Proceedings of the 27th International Conference on International Conference on Machine Learning (ICML)*, Association for Computing Machinery, 2010, pp. 487–494.
[20] M.W. Schmidt, N.L. Roux, and F.R. Bach, Convergence rates of inexact proximal-gradient methods for convex optimization, in NIPS'11 - 25 th Annual Conference on Neural Information Processing Systems, Dec 2011.
[21] F.S. Stonyakin, A. Tyurin, A.V. Gasnikov, P.E. Dvurechensky, A. Agafonov, D. Dvinskikh, M.S. Alkousa, D. Pasechnyuk, S. Artamonov, and V. Piskunova, *Inexact model: a framework for optimization and variational inequalities*. Optim. Methods Softw. 36(6) (2021), pp. 1155–1201.
[22] K.K. Thekumparampil, P. Jain, P. Netrapalli, and S. Oh, *Projection efficient subgradient method and optimal nonsmooth Frank–Wolfe method*, in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, Curran Associates Inc., 2020.
[23] L. Vandenberghe, *Accelerated proximal gradient methods*. https://www.seas.ucla.edu/vandenbe/236C/lectures/fgrad.pdf, 2022. Accessed: 2023–07-07.
[24] S. Villa, S. Salzo, L. Baldassarre, and A. Verri, *Accelerated and inexact forward-backward algorithms*, SIAM J. Optim. 23(3) (2013), pp. 1607–1633.
[25] Y. Xu, *Iteration complexity of inexact augmented Lagrangian methods for constrained convex programming*, Math. Program. 185 (2021), pp. 199–244.
[26] L. Yang and K.-C. Toh, *An inexact Bregman proximal gradient method and its inertial variant*. arXiv e-prints, page arXiv:2109.05690, Sept. 2021.
[27] S. Zhang, J. Yang, C. Guzmán, N. Kiyavash, and N. He, *The complexity of nonconvex-strongly-concave minimax optimization*, in C. de Campos and M. H. Maathuis, editors, *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, PMLR, 27–30 Jul 2021, pp. 482–492.

Appendix A. Miscellaneous proofs

A.1 Proof of Theorem 3.1

***Proof*:** We proceed in two sequential steps: proving (19) then (20) with the help of (19). Let us proceed by *absurdum* by assuming that there exists a pair of indices
$(k_1, k_2) \in \{0, \dots, N-1\}^2$ such that

$$\mathcal{I}_{k_1} \geq \mathcal{I}_{k_2} \wedge \sigma^*_{k_1} < \sigma^*_{k_2}.$$

We consider the function

$$(\sigma_{k_1}, \sigma_{k_2}) \in \mathbb{R}^2 \to f(\sigma_{k_1}, \sigma_{k_2}) := n_{k_1}(\sigma_{k_1}) + n_{k_2}(\sigma_{k_2}) + \sum_{k=0,\, k\neq k_1,\, k\neq k_2}^{N-1} n_k(\sigma^*_k) \in \mathbb{R}_+$$

Trivially, the entries $\sigma^*_{k_1}$ and $\sigma^*_{k_2}$ introduced before form a solution of the two-dimensional problem

$$\min_{\sigma_{k_1}\in\Xi,\ \sigma_{k_2}\in\Xi} f(\sigma_{k_1}, \sigma_{k_2}) \quad \text{s.t. } d_{k_1}(\sigma_{k_1}) + d_{k_2}(\sigma_{k_2}) = \tilde{D} = D - \sum_{k=0,\, k\neq k_1,\, k\neq k_2}^{N-1} d_k(\sigma^*_k).$$

Let us show that $\sigma^*_{k_1} < \sigma^*_{k_2}$ is impossible under the hypotheses above. KKT necessary conditions ensure the existence of $\tilde{\lambda} \in \mathbb{R}$, $\tilde{\mu}_{k_1,\oplus}$, $\tilde{\mu}_{k_1,\ominus} \geq 0$ and $\tilde{\mu}_{k_2,\oplus}$, $\tilde{\mu}_{k_2,\ominus} \geq 0$ with

$$\begin{aligned}
n'_{k_1}(\sigma^*_{k_1}) + \tilde{\lambda}\, d'_{k_1}(\sigma^*_{k_1}) &= (\tilde{\mu}_{k_1,\ominus} - \tilde{\mu}_{k_1,\oplus}) \\
n'_{k_2}(\sigma^*_{k_2}) + \tilde{\lambda}\, d'_{k_2}(\sigma^*_{k_2}) &= (\tilde{\mu}_{k_2,\ominus} - \tilde{\mu}_{k_2,\oplus}) \\
(l - \sigma^*_{k_1}) \cdot \tilde{\mu}_{k_1,\ominus} &= 0 \\
(l - \sigma^*_{k_2}) \cdot \tilde{\mu}_{k_2,\ominus} &= 0 \\
(u - \sigma^*_{k_1}) \cdot \tilde{\mu}_{k_1,\oplus} &= 0 \\
(u - \sigma^*_{k_2}) \cdot \tilde{\mu}_{k_2,\oplus} &= 0.
\end{aligned}$$

The only possible scenarios leading to $l \leq \sigma^*_{k_1} < \sigma^*_{k_2} \leq u$ are achieved when $\tilde{\mu}_{k_1,\oplus} = \tilde{\mu}_{k_2,\ominus} = 0$ implying that

$$-\frac{n'_{k_2}(\sigma^*_{k_2}) + \tilde{\mu}_{k_2,\oplus}}{d'_{k_2}(\sigma^*_{k_2})} = \tilde{\lambda} = -\frac{n'_{k_1}(\sigma^*_{k_1}) - \tilde{\mu}_{k_1,\ominus}}{d'_{k_1}(\sigma^*_{k_1})}. \tag{A1}$$

After elementary calculations, it must follow that

$$\frac{1 + \frac{\tilde{\mu}_{k_2,\oplus}}{n'_{k_2}(\sigma^*_{k_2})}}{1 - \frac{\tilde{\mu}_{k_1,\ominus}}{n'_{k_1}(\sigma^*_{k_1})}} = \frac{\mathcal{I}_{k_1}(\sigma^*_{k_1})}{\mathcal{I}_{k_2}(\sigma^*_{k_2})} < 1.$$

The inequality comes from the fact that $\{\mathcal{I}_k\}_{k=0}^{N-1}$ functions are strictly increasing and strictly negative on Ξ while $\sigma^*_{k_1} < \sigma^*_{k_2}$. Indeed, our inductive step leads to

$$\mathcal{I}_{k_1}(\sigma^*_{k_1}) - \mathcal{I}_{k_2}(\sigma^*_{k_1}) \geq 0 \Rightarrow 0 < \mathcal{I}_{k_1}(\sigma^*_{k_1})/\mathcal{I}_{k_2}(\sigma^*_{k_1}) \leq 1,$$

because of the strict negativity of $\{\mathcal{I}_k\}_{k=0}^{N-1}$. One then notes $\mathcal{I}_{k_2}(\sigma^*_{k_1}) < \mathcal{I}_{k_2}(\sigma^*_{k_2})$.

Therefore, one should observe that

$$0 \leq 1 + \frac{\tilde{\mu}_{k_2,\oplus}}{n'_{k_2}(\sigma^*_{k_2})} < 1 - \frac{\tilde{\mu}_{k_1,\ominus}}{n'_{k_1}(\sigma^*_{k_1})},$$

which turns out to be impossible since $\{n_k\}_{k=0}^{N-1}$ functions are strictly increasing on Ξ.

We conclude that (19) is correct. To prove (20) on the other hand, let us imagine this time that there exists a pair of indices $(k_1, k_2) \in \{0, \ldots, N-1\}^2$ such that

$$\mathcal{I}_{k_1} \succ \mathcal{I}_{k_2} \wedge l < \sigma^*_{k_1} \leq \sigma^*_{k_2} < u.$$

Clearly, (19) already shows that $\sigma^*_{k_1} \not\prec \sigma^*_{k_2}$. The only plausible scenario left resumes in $l < \sigma^*_{k_1} = \sigma^* = \sigma^*_{k_2} < u$. Setting $\tilde{\mu}_{k_1,\oplus} = 0 = \tilde{\mu}_{k_2,\ominus}$ in (A1) yields

$$-\mathcal{I}_{k_2}(\sigma^*) = \tilde{\lambda} = -\mathcal{I}_{k_2}(\sigma^*).$$

Again, this contradicts our assumption that $\mathcal{I}_{k_1} \succ \mathcal{I}_{k_2}$. ■

A.2 Proof of Theorem 3.3

Proof: A neutral schedule $\{\delta_k = \bar{\delta}\}_{k=0}^{N-1}$ certifies, trivially, the feasibility of problem (17) whose basic domain Ξ^N is compact. Since $\bar{\delta} \in \text{int}(\Xi) =]m\bar{\delta}, M\bar{\delta}[$ is fixed, according to the definition of h within Assumption B, we can write

$$0 < \sum_{k=0}^{N-1} b_k\, h(\bar{\delta}) = \bar{\mathcal{B}} < \infty.$$

Before diving into KKT conditions, let us point out two observations:

(1) With $m < 1$, $\{\delta_k = m\bar{\delta}\}_{k=0}^{N-1}$ is clearly not feasible since h is assumed to be strictly decreasing on Ξ.
(2) With $M > 1$, $\{\delta_k = M\bar{\delta}\}_{k=0}^{N-1}$ is clearly not optimal since the neutral schedule above is feasible and
$$\sum_{k=0}^{N-1} a_k\, \bar{\delta} = ||a||_1\, \bar{\delta} < M\, ||a||_1\, \bar{\delta} = \sum_{k=0}^{N-1} a_k\, M\bar{\delta}$$
as $a \succ \mathbf{0}$ according to Assumption (3.1).

Assumptions of Theorem 3.3 satisfy the ones required by Theorem 3.1 so that this latter applies verbatim to problem (17) with $n_k(\cdot) = a_k(\cdot)$ and $d_k(\cdot) = b_k\, h(\cdot)$.

The negative sign of h' ensures the soundness of our ranking (cfr. (21)).

$$\rho(k_1) < \rho(k_2) \Rightarrow \nu_{k_1} = b_{k_1}/a_{k_1} \geq b_{k_2}/a_{k_2} = \nu_{k_2}$$

leading to

$$\mathcal{I}_{k_1}(\cdot) = a_{k_1}/(b_{k_1}\, h'(\cdot)) \succeq a_{k_2}/(b_{k_2}\, h'(\cdot)) = \mathcal{I}_{k_2}(\cdot).$$

Since n'_k/d'_k ratios are strictly increasing and negative, we can claim that there exists $N_\oplus, N_\ominus \in \{0, \ldots, N-1\}$ defining two subsets $\oplus = \{k \,|\, \rho(k) < N_\oplus\}, \ominus = \{k \,|\, \rho(k) > N-1-N_\ominus\}$ of $\{0, \ldots, N-1\}$ such that δ^*_k will be affected to the upper-bound $u = M\bar{\delta}$ if $k \in \oplus$ whereas it will be assigned the lower-bound $l = m\bar{\delta}$ if $k \in \ominus$. By means of our two previous observations,

$$N_\oplus, N_\ominus < N.$$

If $(\oplus, \ominus)$ forms a partition of $\{0, \ldots, N-1\}$ with

$$\sum_{k\in\oplus} b_k\, h(M\bar{\delta}) + \sum_{k\in\ominus} b_k\, h(m\bar{\delta}) = \bar{\mathcal{B}}, \tag{A2}$$

then the inherent schedule $\delta_k = M\bar{\delta}$ for every $k \in \oplus$, $\delta_k = m\bar{\delta}$ for every $k \in \ominus$ might be a solution of (17). The structure of such schedule is identical to the one expected by (23). We consider now other possible solutions. KKT necessary conditions (see LICQ below) imply that if δ^* is a local optimum of (17), then there exist $\tilde{\lambda} \in \mathbb{R}$, $\tilde{\mu}_\ominus$, $\tilde{\mu}_\oplus \in \mathbb{R}^N_+$ (dual feasibility) such that equalities and equalities of Table A1 are jointly satisfied. When the context is not misleading, we make the abuse of notation for any $\delta \in \Xi^N$, $h'(\delta) = (h'(\delta_0), \ldots, h'(\delta_{N-1}))^T$ and $h(\delta) = (h(\delta_0), \ldots, h(\delta_{N-1}))^T$.

■

Indeed, (17) enjoys the LICQ constraint qualification at any $\delta \in \Xi^N$ not matching the degenerate case (A2). At any feasible $\delta \in \Xi^N$, the span of active constraints' gradients always involve $h'(\delta)$ whose entries are all non-zero, excepted eventually at $k_+ \in \oplus$ where $\delta_{k_+} = M\bar{\delta}$ if $h'(M\bar{\delta}) = 0$ (due to the convexity of h and strict monotony of h'). Gradients of other active constraints are vectors $\mathbf{e}_{k_+}$ for $k_+ \in \oplus$ and $\mathbf{e}_{k_-}$ for $k_- \in \ominus$ that are all linearly independent. Since we explicitly discard schedules of (A2) type, it must exist an entry $k \in \{0, \dots, N-1\}$ where bound constraints are inactive, hence $h'(\delta_k) < 0$. To obtain the $\mathbf{0}$ vector from a linear combination of active constraints' gradients, one should then take zero times $h'(\delta)$. If no bound constraint is active at δ then the induced linear combination immediately turns out to be the trivial one. Conversely, if some bound constraints are active, then their associated coefficients in the linear combination must also be zero as shown previously, giving rise again to the null combination.

Note that KKT conditions are sufficient since the relaxation (convex) of (17)

$$\min_{m\bar{\delta} \preceq \delta \preceq M\bar{\delta}} \sum_{k=0}^{N-1} a_k \, \delta_k \quad \text{s.t.} \sum_{k=0}^{N-1} b_k \, h(\delta_k) \leq \bar{\mathcal{B}} \tag{A3}$$

would always lead to an optimal solution that tightens the budget constraint as $a \succ \mathbf{0}$ and h is strictly decreasing. For writing ease, let us now denote

$$\mathcal{T} = \{0, \dots, N-1\} \backslash (\oplus \cup \ominus)$$

In line with the comments made so far, we look after a schedule $\{\delta_k^*\}_{k=0}^{N-1}$ of the form

$$\begin{cases} \delta_k^* = M\,\bar{\delta} & k \in \oplus \\ \delta_k^* \in]m\bar{\delta}, M\bar{\delta}[& k \in \mathcal{T} \\ \delta_k^* = m\,\bar{\delta} & k \in \ominus \end{cases} \tag{A4}$$

for two positive integers $N_\oplus, N_\ominus$ with $0 \leq N_\oplus + N_\ominus \leq N-1$.

For any $k \in \mathcal{T}$, stationarity requirement compels

$$a_k + \tilde{\lambda}\, b_k \, h'(\delta_k^*) = 0 \Rightarrow h'(\delta_k^*) = -(\tilde{\lambda}\, a_k^{-1}\, b_k)^{-1} = \frac{a_k\, \lambda^*}{b_k} \Rightarrow \delta_k^* = (h')^{(-1)} \left(\frac{a_k\, \lambda^*}{b_k} \right)$$

for $\lambda^* = -(\tilde{\lambda})^{-1} \in \mathbb{R}$. Primal feasibility (budget) row of Table A1 translates to (24).

A.3 Uniqueness

Proof: We prove here the uniqueness of δ^*. For ease of writing, let us reuse the notation

$$\sum_{k=0}^{N-1} b_k \, h(\bar{\delta}) = \bar{\mathcal{B}}.$$

Aside from the fact that all the entries ν_k of ν supposedly differ, Assumption 3.1 about h certifies that for any fixed $N_\oplus, N_\ominus$, there can be at most one $\lambda^* \in \mathbb{R}$ ensuring (24). Seen from a different angle, for any couple $(N_\oplus, N_\ominus)$ there can be a single $\{\delta_k^*(N_\oplus, N_\ominus)\}_{k=0}^{N-1}$ respecting both (23) and (24). Let

Table A1. KKT conditions for (17).

$a + \tilde{\lambda}\, b \odot h'(\delta^*) = (\tilde{\mu}_\ominus - \tilde{\mu}_\oplus)$	Stationarity
$m\bar{\delta} \preceq \delta^* \preceq M\bar{\delta}$	Primal feasibility (bounds)
$b^T h(\delta^*) = \bar{\mathcal{B}}$	Primal feasibility (budget)
$(m\bar{\delta} - \delta^*) \odot \tilde{\mu}_\ominus = \mathbf{0}$	Complementary (–)
$(\delta^* - M\bar{\delta}) \odot \tilde{\mu}_\oplus = \mathbf{0}$	Complementary (+)

us assume by *absurdum* the optimality of two such schedules. Then, there must exist two couples $(N_{\oplus,1}, N_{\ominus,1})$ and $(N_{\oplus,2}, N_{\ominus,2})$ such that

$$\sum_{k=0}^{N-1} a_k\, \delta_k^*(N_{\oplus,1}, N_{\ominus,1}) = u^* = \sum_{k=0}^{N-1} a_k\, \delta_k^*(N_{\oplus,2}, N_{\ominus,2}).$$

Any schedule $\hat{\delta}^{(t)}$ *in between* for $t \in]0,1[$ with

$$\hat{\delta}_k^{(t)} = t \cdot \delta_k^*(N_{\oplus,1}, N_{\ominus,1}) + (1-t) \cdot \delta_k^*(N_{\oplus,2}, N_{\ominus,2})$$

for every $k \in \{0, \dots, N-1\}$ would trivially also yield the optimal cost u^*. By convexity of the set $\mathcal{H} \subset \Xi^N$,

$$\mathcal{H} = \left\{ \delta \in \Xi^N \mid \sum_{k=0}^{N-1} b_k\, h(\delta_k) \leq \bar{\mathcal{B}} \right\},$$

we have that $\delta^*(N_{\oplus,1}, N_{\ominus,1}) \in \mathcal{H} \wedge \delta^*(N_{\oplus,2}, N_{\ominus,2}) \in \mathcal{H} \Rightarrow \hat{\delta}^{(t)} \in \mathcal{H}$. We examine two mutually exclusive situations.

(1) $\sum_{k=0}^{N-1} b_k\, h(\hat{\delta}_k^{(t)}) < \bar{\mathcal{B}}$. Then, it is possible to decrease at least one entry of $\hat{\delta}^{(t)}$ and obtain a strictly better objective value meaning that $\delta^*(N_{\oplus,1}, N_{\ominus,1})$ and $\delta^*(N_{\oplus,2}, N_{\ominus,2})$ were not jointly optimal: contradiction.

(2) $\sum_{k=0}^{N-1} b_k\, h(\hat{\delta}_k^{(t)}) = \bar{\mathcal{B}}$. In such scenario, the entire segment $[\delta^*(N_{\oplus,1}, N_{\ominus,1}), \delta^*(N_{\oplus,2}, N_{\ominus,2})]$ must be an edge of $\mathcal{H}$. We can quickly invoke continuity arguments suggesting the existence of multiple schedules on that segment sharing both a common number of $\delta_k^* = M\bar{\delta}$ (same $N_\oplus$ value) and a common number of $\delta_k^* = m\bar{\delta}$ (same $N_\ominus$ value). Yet by linearity their objective value would remain equal to u^*. As mentioned above, this is impossible because there exists at most one valid schedule $\delta^*(N_\oplus, N_\ominus)$ by pair $(N_\oplus, N_\ominus)$: contradiction.

■

A.4 Closed-form schedules with cut-off

We extend the family of $\{h_{r>0}\}$ functions and incorporate the limiting case $r = 0$.

For any $r \geq 0$,

$$h_r(\delta) = \begin{cases} -\log(\delta) & r = 0 \\ (\delta)^{-r} & r > 0 \end{cases} \Rightarrow (h_r')^{(-1)}(-\omega) = \begin{cases} (\omega)^{-1} & r = 0 \\ \left(\frac{\omega}{r}\right)^{-\frac{1}{r+1}} & r > 0. \end{cases} \tag{A5}$$

We kindly remind that we only focus on instances of h functions positive on their respective domains. Both Theorems A.1 and A.2 immediately follow from Theorem 3.3, their respective proof is therefore omitted.

Theorem A.1: *Let Assumptions* A *and* B *hold with* $(a,b) \in \mathbb{R}_{++}^{N\times 2}$, $m < 1 < M$ *and* $h = h_{r\geq 0} : [m\,\bar{\delta}, M\,\bar{\delta}] \to \mathbb{R}_+$.

$\exists N_\oplus, N_\ominus \in \{0, \dots, N-1\}$, $\hat{\lambda} \in \mathbb{R}$ *such that* $\forall\, k \in \{0, \dots, N-1\}$

$$\delta_k^* = \begin{cases} M\,\bar{\delta} & k \in \oplus := \{\tilde{k} \mid \rho(\tilde{k}) < N_\oplus\} \\ \hat{\lambda}\left(\frac{b_k}{a_k}\right)^{\frac{1}{r+1}} & k \in \mathcal{T} := \{\tilde{k} \mid N_\oplus \leq \rho(\tilde{k}) \leq N-1-N_\ominus\} \\ m\,\bar{\delta} & k \in \ominus := \{\tilde{k} \mid \rho(\tilde{k}) > N-1-N_\ominus\}, \end{cases} \tag{A6}$$

where $\rho(k)$ *depicts the descending rank of* $v_k = b_k/a_k$, $\hat{\lambda} > 0$ *is defined as*

$$\hat{\lambda} = \begin{cases} \bar{\delta} \cdot \left[\frac{M^{\sum_{k\in\oplus} b_k} . m^{\sum_{k\in\ominus} b_k}}{\Pi_{k\in\mathcal{T}} (b_k/a_k)^{-b_k}} \right]^{-\left(\sum_{k\in\mathcal{T}} b_k\right)^{-1}} & r = 0 \\ \bar{\delta} \cdot \left[\frac{\left(\sum_{k=0}^{N-1} b_k\right) - \left(\sum_{k\in\oplus} b_k\right) M^{-r} - \left(\sum_{k\in\ominus} b_k\right) m^{-r}}{\sum_{k\in\mathcal{T}} \left(a_k^r\, b_k\right)^{\frac{1}{r+1}}} \right]^{-\frac{1}{r}} & r > 0 \end{cases} \tag{A7}$$

and δ^* *stands as a solution of* (17).

Theorem A.2: *Let Assumptions* A *and* B *hold with* $(a, b) \in \mathbb{R}_{++}^{N\times 2}$, $\omega_M < \bar{\omega}/N < \omega_m$ *and* $h = h_{r>0} : \mathbb{R}_+ \to \mathbb{R}_+$.

$\exists N_\oplus, N_\ominus \in \{0, \ldots, N-1\}$, $\hat{\lambda} \in \mathbb{R}$ *such that* $\forall\, k \in \{0, \ldots, N-1\}$

$$\omega_k^* = \begin{cases} \omega_M & k \in \oplus := \{\tilde{k} \,|\, \rho(\tilde{k}) < N_\oplus\} \\ \hat{\lambda}\, (a_k^r\, b_k)^{\frac{1}{r+1}} & k \in \mathcal{T} := \{\tilde{k} \,|\, N_\oplus \le \rho(\tilde{k}) \le N - 1 - N_\ominus\} \\ \omega_m & k \in \ominus := \{\tilde{k} \,|\, \rho(\tilde{k}) > N - 1 - N_\ominus\}, \end{cases} \tag{A8}$$

where $\rho(k)$ *depicts the descending rank of* $v_k = (a_k^r\, b_k)^{-1}$, $\hat{\lambda} > 0$ *is defined as*

$$\hat{\lambda} = \left[\frac{\bar{\omega} - N_\oplus\, \omega_M - N_\ominus\, \omega_m}{\sum_{k\in\mathcal{T}} \left(a_k^r\, b_k\right)^{\frac{1}{r+1}}} \right] \tag{A9}$$

and ω^* *stands as a solution of* (30).