A Universal Transfer Theorem for Convex Optimization Algorithms Using Inexact First-order Oracles

Anonymous Authors[1]

Abstract

Given *any* algorithm for convex optimization that uses exact first-order information (i.e., function values and subgradients), we show how to use such an algorithm to solve the problem with access to *inexact* first-order information. This is done in a "black-box" manner without knowledge of the internal workings of the algorithm. This complements work done by Devolder-Glineur-Nesterov and Schmidt-Le Roux-Bach who consider the performance of specific algorithms like (accelerated) gradient descent with inexact information. In particular, our results apply to a wider range of algorithms beyond variants of gradient descent, e.g., projection-free methods, cutting-plane methods, or any other first-order methods formulated in the future. Further, they also apply to algorithms that handle structured nonconvexities like mixed-integer decision variables.

1. Introduction

Optimization is a core tool for almost any learning or estimation problem. Such problems are very often approached by setting up an optimization problem whose decision variables model the entity to be estimated, and whose objective and constraints are defined by the observed data combined with structural insights into the inference problem. Algorithms for any sufficiently general class of relevant optimization problems in such settings need to collect information about the particular instance by making (adaptive) queries about the objective before they can report a good solution. In this paper, we focus on the following important class of optimization problems *over a fixed ground set* $X \subseteq \mathbb{R}^d$

$$\min\{f(x) : x \in X\}, \tag{1.1}$$

where $f : \mathbb{R}^d \to \mathbb{R}$ is a (possibly nonsmooth) convex function. When the underlying ground set X is all of $\mathbb{R}^d$ or some fixed convex subset, (1.1) is the classical convex optimization problem. In this paper, we allow X to be more general and to be used to model some known nonconvexity, e.g. integrality constraints by setting $X = C \cap (\mathbb{Z}^{d_1} \times \mathbb{R}^{d_2})$ with $d_1 + d_2 = d$, where $C \subseteq \mathbb{R}^d$ is a fixed convex set. From an algorithmic perspective, the setup is that the algorithm has complete knowledge of what X is, but does not *a priori* know f and must collect information via queries. A standard model for accessing the function f is through so-called *first-order oracles*. At any point during its execution, the algorithm can request the function value and the (sub)gradient of f at any point $\bar{x} \in \mathbb{R}^d$.

Given access to such oracles, a *first-order algorithm* makes adaptive queries to this oracle and, after it judges that it has collected enough information about f, it reports a solution with certain guarantees. A long line of research has gone into understanding exactly how many queries are needed to solve different classes of problems (with different sets of assumptions on f and X), with tight upper and lower bounds on the query complexity (a.k.a. *oracle* or *information complexity*) known in the literature; see (Nesterov, 2004; Bubeck, 2015; Nemirovski, 1994; Basu, 2023; Basu et al., 2023) for expositions of these results.

A natural question that arises in this context is what happens if the response of the oracle is not exact, but approximate (with possibly desired accuracy). For example, the response of the oracle might be itself a solution to another computational problem which is solved only approximately, which happens when using function smoothing (Nesterov, 2005), and in minimax problems (Wang & Abernethy, 2018). Stochastic first-order oracles, modeling applications where only some estimate of the gradient is used, may also be viewed as inexact oracles whose accuracy is a random variable at each iteration. Thus, researchers have also investigated what one can say about algorithms that have access to inexact oracle responses (with possibly known guarantees on the inexactness). Early work on this topic appears in Shor (Shor, 1985) and Polyak (Polyak, 1987), and more recent progress can be found in (Devolder et al., 2014; Schmidt et al., 2011; Lan, 2009; Hintermüller, 2001; Kiwiel,

[1] Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

2006; d'Aspremont, 2008) and references therein. To the best of our knowledge, all previous work on inexact first-order oracles has focused either on how specific algorithms like (accelerated) gradient methods perform with inexact (sub)gradients with no essential change to the algorithm, or on how to adapt a particular class of algorithms to perform well with inexact information.

In this paper, we provide a different approach to the problem of inexact information. We provide a way to take *any* first-order algorithm that solves (1.1) with exact first-order information and, with absolutely no knowledge of the its inner workings, show how to make the same algorithm work with inexact oracle information. Thus, in contrast to earlier work, our result is not the analysis of specific algorithms under inexact information or the adaptation of specific algorithms to use inexact information. It is in this sense that we believe our results to be *universal* because they apply to a much wider class of algorithms than previous work, including gradient descent, cutting plane methods, bundle methods, projection-free methods etc., and also to any first-order method that is invented in the future for optimization problems of the form (1.1).

2. Formal statement of results and discussion

We begin with definitions of standard concepts that we need to state our results formally. We use $\|\cdot\|$ to denote the standard Euclidean norm and $B(c,r)$ to denote the Euclidean ball of radius r centered at $c \in \mathbb{R}^d$. When the center is the origin, we denote the ball by $B(r)$. A function $h : \mathbb{R}^d \to \mathbb{R}$ is said to be M-Lipschitz if $|h(x) - h(y)| \leq M\|x - y\|$ for all x, y. We now formalize the inexact first-order oracles that we will work with.

Definition 2.1. An *η-approximate first-order oracle* for a convex function $f : \mathbb{R}^d \to \mathbb{R}$ takes as input a query point $\bar{x} \in \mathbb{R}^d$ and returns a first-order pair $(\tilde{f}, \tilde{g})$ satisfying $|\tilde{f} - f(\bar{x})| \leq \eta$ and $\|\tilde{g} - g\| \leq \frac{\eta}{2R}$ for some subgradient $g \in \partial f(\bar{x})$.

We now state our main results. Let $\mathcal{F}^0(M,R)$ denote the standard family of instances of the optimization problem (1.1) consisting of all M-Lipschitz (possibly nondifferentiable) convex functions f such that the minimizer $x^\star \in X$ is contained in the ball $B(R)$[1]. We remind the reader that the underlying set X need not be $\mathbb{R}^d$ and may be nonconvex; below, when we talk about a first order algorithm for (1.1) we mean an algorithm that can solve (1.1) with access to first-order oracles for f. We use $\text{OPT}(f)$ to denote the optimal value of the instance f.

Theorem 2.2. Consider an algorithm for (1.1) such that for any instance $f \in \mathcal{F}^0(M,R)$, with access to function values and subgradients of f, after T iterations the algorithm reports a feasible solution $x \in X$ with error at most $err(T,M,R)$, i.e., $f(x) \leq \text{OPT}(f) + err(T,M,R)$.

Then there is an algorithm that, with access to an η-approximate first-order oracle for f for any $\eta \geq 0$, after T iterations the algorithm returns a feasible solution $\bar{x} \in X$ with value

$$f(\bar{x}) \leq \text{OPT}(f) + err(T, M', R) + 4\eta T,$$

where $M' = M + \frac{\eta}{2R}$.

Though we state this theorem as an existence result, our proof is constructive and exactly formulates the desired algorithm via Procedures 1 and 2. Let us illustrate what this theorem says when applied to two classical algorithms for convex optimization (i.e., $X = \mathbb{R}^d$): subgradient methods and cutting-plane methods (Nesterov, 2004). When using *exact* first-order information, the subgradient method produces after T iterations a solution with error at most $O\left(\frac{MR}{\sqrt{T}}\right)$. Applying the procedures mentioned from Theorem 2.2 to this algorithm, one obtains an algorithm that uses only η-approximate first-order information and after T iterations produces a solution whose error is at most $O\left(\frac{MR+\eta}{\sqrt{T}}\right) + 2T\eta$. If one can choose the accuracy of the inexact oracle, setting $\eta = \frac{\varepsilon^3}{M^2R^2}$ and $T = \lceil\frac{M^2R^2}{\varepsilon^2}\rceil$ gives a solution with error at most $O(\varepsilon)$. Note that this does not involve knowing anything about the original algorithm; it simply illustrates the tradeoff between the oracle accuracy and final solution accuracy.

Similarly, for classical cutting-plane methods (e.g., center-of-gravity, ellipsoid, Vaidya) the error after $T \geq \text{poly}(d)\log\left(\frac{R}{\rho}\right)$ iterations is at most $\left(\frac{MR^2}{\rho}\right)\exp\left(\frac{-T}{\text{poly}(d)}\right)$. Thus, with access to η-approximate first-order oracles, we can use our result to produce a solution with error at most $\left(\frac{(M+\frac{\eta}{2R})R^2}{\rho}\right)\exp\left(\frac{-T}{\text{poly}(d)}\right) + 4\eta T$. With the desired accuracy of $\eta = \frac{\varepsilon}{\text{poly}(d)\log\left(\frac{MR}{\rho\varepsilon}\right)}$, and $T = 2\,\text{poly}(d)\log\left(\frac{MR}{\rho\varepsilon}\right)$, it gives a solution with error at most $O(\varepsilon)$.

We next consider the family of *α-smooth functions*, i.e., the family $\mathcal{F}^1(M,\alpha,R)$ of M-Lipschitz convex functions that are differentiable with α-Lipschitz gradient maps, whose minimizers are contained in $B(R)$. This is a classical family of objective functions in convex optimization that admits

[1] This is a standard assumption in the analysis of optimization algorithm – if no such bound is assumed, then it can be shown that no algorithm can report a good solution within a guaranteed number of steps for every instance (Nesterov, 2004). Alternatively, one may give the convergence rates in terms of the distance of the initial iterate of the algorithm and the optimal solution (one can think of R as an upper bound on this distance). Our results can also be formulated in this language with no conceptual or technical changes.

the celebrated accelerated method of Nesterov (1983) (see (d'Aspremont et al., 2021) for a survey). We give the following universal transfer theorem for algorithms for smooth objective functions.

Theorem 2.3. Consider an algorithm for (1.1) such that for any instance in $f \in \mathcal{F}^1(M, \alpha, R)$, with access to function values and subgradients of f, after T iterations the algorithm reports a feasible solution $x \in X$ with error at most $err(T, M, \alpha, R)$, i.e., $f(x) \leq \text{OPT}(f) + err(T, M, \alpha, R)$.

Then there is an algorithm that, with access to an η-approximate first-order oracle for f, after T iterations the algorithm returns a feasible solution $\bar{x} \in X$ with value

$$f(\bar{x}) \;\leq\; \text{OPT}(f) + err(T, M', \alpha', R) + 5\eta \cdot (T+2),$$

where $M' = M + \frac{\eta}{2R}$, $\alpha' = \alpha \cdot \sqrt{d} \cdot \left(4\sqrt{5(T+1)} + 3\right)$.

As an illustration, we apply this transfer theorem to the accelerated algorithm of Nesterov (1983) for continuous optimization ($X = \mathbb{R}^d$): Under *perfect* first-order information, it obtains error $O(\frac{\alpha R^2}{T^2})$ after T iterations. Using our transfer theorem as a wrapper gives an algorithm that, using only η-approximate first-order information, obtains error $O(\frac{\alpha\sqrt{d}}{T^{1.5}} + \eta T)$; if the accuracy of the oracle is set to $\eta = O(\frac{\alpha\sqrt{d}}{T^{2.5}})$, this gives an algorithm with error $O(\frac{\alpha\sqrt{d}}{T^{1.5}})$. While this does not recover in full the acceleration of Nesterov's method, the key take away is that a significant amount of acceleration (i.e., error rates better than those possible for non-smooth functions) can be preserved under inexact oracles in a *universal* way, for *any* accelerated algorithm requiring exact information.

Remark 2.4. For the sake of exposition, we have assumed that the accuracy η of the oracle is fixed and the additional error is $O(\eta T)$. However, one can allow different oracle accuracies η_t at each query point x_t and the additional error is $O(\sum_t \eta_t)$ (and the parameter $M' = M + \frac{1}{2R} \max_t \eta_t$).

2.1. Allowing inexactness in the constraint set

So far we have assumed that the algorithm has complete knowledge of the constraints X. Now, we extend our results to include algorithms that can work with larger classes of constraints that are not fully known up front. In other words, just like the algorithm needs to collect information about f, it also needs to collect information about X, via another oracle, to be able to solve the problem. To capture the most general algorithms of this type, we formalize this setting by assuming X is of the form $C \cap Z$, where C belongs to a class of closed, convex sets and Z is possibly nonconvex but completely known (e.g., $Z = \mathbb{Z}^{d_1} \times \mathbb{R}^{d_2}$ with $d_1 + d_2 = d$).

$$\min\{f(x) : x \in C \cap Z\}. \tag{2.1}$$

The algorithm then must collect information about C, for which we use the common model of allowing the algorithm access to a separation oracle. Upon receiving a query point x, a separation oracle either reports correctly that x is inside C or otherwise returns a separating hyperplane that separates x from C. We note that a separation oracle for C is in some sense comparable to a first-order oracle for a convex function f; since the pair $(f(x), \nabla f(x))$ can be viewed as providing a supporting hyperplane for the epigraph of f at x, using an oracle that returns separating hyperplanes for C provides a comparable way of collecting information about the constraints. Let us first precisely define the inexact version of a separation oracle.

Definition 2.5. For a closed, convex set $C \subseteq B(R)$ and a query point $\bar{x} \in B(R)$, an *η-approximate separation oracle* reports a separation response $(flag, \tilde{g}) \in \{\textsc{Feasible}, \textsc{Infeasible}\} \times \mathbb{R}^d$ such that if $\bar{x} \in C$ then $flag = \textsc{Feasible}$ (with no requirement on $\tilde{g}$), and otherwise $flag = \textsc{Infeasible}$ and $\tilde{g}$ is a unit vector such that there exists some unit vector g satisfying $\langle g, x\rangle \leq \langle g, \bar{x}\rangle$ for all $x \in C$ and $\|\tilde{g} - g\|_2 \leq \frac{\eta}{4R}$. Given such a $\tilde{g}$ (for $\bar{x} \notin C$), we call the hyperplane through $\bar{x}$ induced by this normal vector an *η-approximate separating hyperplane* for $\bar{x}$.

We now state our results for algorithms that work with separation oracles. Note that for this, instances of (1.1) have to specify both f and C, as opposed to just f, since only Z is known but not C. We use $\mathcal{I}(M, R, \rho)$ to denote the set of all instances (f, C) where $f : \mathbb{R}^d \to \mathbb{R}$ is an M-Lipschitz convex function and C is a compact, convex set that contains a ball of radius ρ and is contained in $B(R)$. We use $OPT(f, C)$ to denote the minimum value of (2.1). The "strict feasibility" assumption of C containing a ρ-ball is standard in convex optimization with constraints given via separation oracles. Otherwise, it can be shown that no algorithm will be able to find even an approximately feasible point in a finite number of steps (Nesterov, 2004). The first result we state is for pure convex problems, i.e., $Z = \mathbb{R}^d$.

Theorem 2.6. Consider an algorithm for (2.1) with $Z = \mathbb{R}^d$, such that for any instance in $(f, C) \in \mathcal{I}(M, R, \rho)$, with access to function values and subgradients of f and separating hyperplanes for C, after T iterations the algorithm reports a feasible solution $x \in C$ with error at most $err(T, M, R, \rho)$, i.e., $f(x) \leq \text{OPT}(f, C) + err(T, M, R, \rho)$.

Then there is an algorithm that, with access to an η_f-approximate first-order oracle for f and an η_C-approximate separation oracle for C for any $\eta_f \geq 0$ and $0 \leq \eta_C \leq \rho$, after T iterations the algorithm returns a feasible solution $\bar{x} \in C$ with value

$$f(\bar{x}) \;\leq\; \text{OPT}(f, C) + err(T, M', R, \rho') + 4\eta_f T + \frac{2\eta_C M R}{\rho},$$

where $M' = M + \frac{\eta_f}{2R}$ and $\rho' = \rho - \eta_C$.

We can handle more general, nonconvex Z with separation oracles under a slightly stronger “strict feasibility" assumption on C: let $\mathcal{I}^\star(M, R, \rho)$ denote the subclass of instances from $\mathcal{I}(M, R, \rho)$ where the minimizer $x^\star$ of (2.1) is ρ-deep inside C, i.e., $B(x^\star, \rho) \subseteq C$.

Theorem 2.7. Consider an algorithm for (2.1), such that for any instance in $(f, C) \in \mathcal{I}^\star(M, R, \rho)$, with access to function values and subgradients of f and separating hyperplanes for C, after T iterations the algorithm reports a feasible solution $x \in C \cap Z$ with error at most $err(T, M, R, \rho)$, i.e., $f(x) \leq \text{OPT}(f, C) + err(T, M, R, \rho)$.

Then there is an algorithm that, with access to an η_f-approximate first-order oracle for f and an η_C-approximate separation oracle for C for any $\eta_f \geq 0$ and $0 \leq \eta_C \leq \rho$, after T iterations the algorithm returns a feasible solution $\bar{x} \in C \cap Z$ with value

$$f(\bar{x}) \leq \text{OPT}(f, C) + err(T, M', R, \rho') + 4\eta_f T,$$

where $M' = M + \frac{\eta_f}{2R}$ and $\rho' = \rho - \eta_C$.

Remark 2.8. The objective functions in the above results were allowed to be any M-Lipschitz, possibly nondifferentiable, convex function. One can state versions of these results for algorithms that work for the smaller class of α-smooth functions (e.g., accelerated projected gradient methods), just as Theorem 2.3 is a version of Theorem 2.2 for α-smooth objectives. The reason is that the analysis for handling constraints is independent of the arguments needed to handle the objective using inexact oracles; however, for space constraints, we leave the details out of this manuscript. Additionally, one can prove versions of all our theorems for strongly convex objective functions, but we leave these out of the manuscript as well to convey the main message of the paper more crisply.

2.2. Relation to existing work

Previous work on inexact first-order information focused on how certain known algorithms perform or can be made to perform under inexact information, most recently on (accelerated) proximal-gradient methods. For instance, (Devolder et al., 2014) analyze the performance of (accelerated) gradient descent in the presence of inexact oracles, with no change to algorithm. They show that simple gradient descent (for unconstrained problems) will return a solution with additional error $O(\eta)$ and accelerated gradient descent encurs an additional error of $O(\eta T)$ (similar to our guarantees). Similarly, (Schmidt et al., 2011) does an analysis for (accelerated) proximal gradient methods, with more complicated forms of the additional error, depending on how well the proximal problems are solved.

In contrast, our result does not assume any knowledge of the internal logic of the algorithm. We must, therefore, use the algorithm in a “black-box” manner. We are able to do this by using the inexact oracles to construct a modified instance whose optimal solution is similar in quality to that of the true instance, and where this inexact information from the true instance can be interpreted as *exact* information for the modified instance. Thus, we can effectively run the algorithm as a black-box on this modified instance and leverage its error guarantee. Constructing this modified instance in an online fashion requires technical ideas that are new, to the best of our knowledge, in this literature. For instance, it is not even true that given approximate function values and subgradients of a convex function, we can find another convex function that has these as exact function values and subgradients; see Figure 1. Thus, one cannot directly use the inexact information as is (contrary to what is done in many of the papers dealing with inexact information for *specific* algorithms), in the general case we consider. The key is to modify the inexact information so that the information the algorithm receives admits an *extension into a convex function/set* that is still close to the original instance. When dealing with α-smooth objectives, the arguments are especially technically challenging since we have to report approximate function and gradient values that allow for a *smooth* extension that also approximates the unknown objective well. This involves careful use of new, localized smoothing techniques and maximal couplings of probability distributions. Such smoothing guarantees based on the proximity to the class of smooth functions may be of independent interest (see Theorem 4.1).

New applications: Since our results apply to algorithms for any ground set X, we are able to handle *mixed-integer convex optimization*, i.e., $X = C \cap (\mathbb{Z}^{d_1} \times \mathbb{R}^{d_2})$, with inexact oracles. Recently, there have been several applications of such optimization problems in machine learning and statistics (Bertsimas et al., 2016; Mazumder & Radchenko, 2017; Bandi et al., 2019; Dedieu et al., 2021; Dey et al., 2022; Hazimeh et al., 2022; 2023). General algorithms for mixed-integer convex optimization, as well as specialized ones designed for specific applications in the above papers, all involve a sophisticated combination of techniques like branch-and-bound, cutting planes and other heuristics. To the best of our knowledge, the performance of these algorithms has never been analyzed under the presence of inexact oracles which can cause issues for all of these components of the algorithm. Our results apply immediately to all these algorithms, precisely because the internal workings of the algorithm are abstracted away in our analysis. This yields the first ever versions of these methods that can work with inexact oracles. Moreover, X can be used to model other types of structured nonconvexities (e.g., complementarity constraints (Cottle et al., 2009)) and our results show how to adapt algorithms in those settings to work with inexact oracles. Note that this holds for the cases where the convex set C is explicitly known a priori (Theorems 2.2 and 2.3), or must be accessed via separation oracles (Theorems 2.6 and 2.7).

The remainder of this paper is dedicated to the proof sketches of Theorems 2.2 and 2.3. The missing details and proofs of Theorems 2.6 and 2.7 can be found in the Appendix.

3. Universal transfer for Lipschitz functions

In this section we prove our transfer result stated in Theorem 2.2. The proof relies on the following key concept: Given a set of points $x_1, \ldots, x_T \in B(R)$ (e.g., queries made by an optimization algorithm), we say that the sequence of first-order pairs[2] $(f_1, g_1), \ldots, (f_T, g_T) \in \mathbb{R} \times \mathbb{R}^d$ has an *M-Lipschitz convex extension*, or simply *M-extension*, if there is a function F that is convex, M-Lipschitz, and such that $f_t = F(x_t)$ and $g_t \in \partial F(x_t)$ for all t, i.e., the first-order information of F at the queried points is **exactly** $\{(f_t, g_t)\}_t$.

As mentioned in the introduction, the main idea is to feed to the convex optimization algorithm $\mathcal{A}$ a sequence of pairs (f_t, g_t)'s that have an M-Lipschitz extension F that is close to the original function f. Since the information is consistent with what the algorithm expects when interacting exactly with the function F, it will approximately optimize the latter which will then give an approximately optimal solution to the neighboring function f.

Unfortunately, it is easy to see approximate first-order information from f for the queried points x_t's does not necessarily have a Lipschitz convex extension (see Figure 1). Thus, the main subroutine of our algorithm *Approximate-to-Exact* given below is that given an approximate first-order oracle for f, it constructs first-order pairs $(\hat{f}_t, \hat{g}_t)$'s in an online fashion (i.e. $(\hat{f}_t, \hat{g}_t)$ only depends on $x_1, \ldots, x_t$) with the desired extension properties.

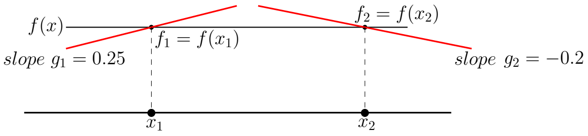

Figure 1: An example where two approximate function values and subgradients do not have a convex extension. The true function f is constant. The function values are reported with no error. The reported slopes are shown in red. However, these slopes decrease going from x_1 to x_2 thus eliminating the possibility of any convex function having these values and slopes at x_1 and x_2.

Theorem 3.1 (Online first-order Lipschitz-extensibility)**.** Consider an M-Lipschitz convex function $f : B(R) \to \mathbb{R}$, and a sequence of points $x_1, \ldots, x_T \in B(R)$. There is an online procedure that, given η-approximate first-order oracle access to f, produces first-order pairs $(\hat{f}_1, \hat{g}_1), \ldots, (\hat{f}_T, \hat{g}_T)$ that have a $(M + \frac{\eta}{2R})$-extension $F : B(R) \to \mathbb{R}$ satisfying $\|F - f\|_\infty \leq 2\eta T$. (Moreover, the procedure only probes the approximate oracle at the given points $x_1, \ldots, x_T$.)

With this at hand, given any first-order algorithm $\mathcal{A}$ we can run it using only approximate first-order information in the following natural way:

Procedure 1. Approximate-to-Exact($\mathcal{A}$, T)

For each timestep $t = 1, \ldots, T$:

1. Receive query point $x_t \in B(R)$ from $\mathcal{A}$.
2. Send point x_t to the η-approximate oracle for f and receive the information $(\tilde{f}_t, \tilde{g}_t)$.
3. Use the online procedure from Theorem 3.1 to construct the first-order pair $(\hat{f}_t, \hat{g}_t)$.
4. Send $(\hat{f}_t, \hat{g}_t)$ to the algorithm $\mathcal{A}$.

Return the point in X returned by $\mathcal{A}$.

Proof of Theorem 2.2. Consider a first-order algorithm $\mathcal{A}$ that, for any M-Lipschitz convex function, after T iterations returns a point $\bar{x} \in X$ such that $f(\bar{x}) \leq \mathrm{OPT}(f) + err(T, M, R)$, We show that running Procedure 1 with $\mathcal{A}$ as input, which only uses an η-approximate oracle for f, returns a point $\bar{x} \in X$ such that $f(\bar{x}) \leq \mathrm{OPT}(f) + err(T, M', R) + 4\eta T$ with $M' = M + \frac{\eta}{2R}$.

To see this, let F be an M'-extension for the first-order pairs $(\hat{f}_t, \hat{g}_t)$ sent to the algorithm $\mathcal{A}$ in Procedure 1 with $\|F - f\|_\infty \leq 2\eta T$, guaranteed by Theorem 3.1. This means that the execution of the first-order algorithm $\mathcal{A}$ during our procedure is exactly the same as executing $\mathcal{A}$ directly on the convex function F. Thus, by the error guarantee of $\mathcal{A}$, the point $\bar{x} \in X$ returned by $\mathcal{A}$ after T iterations (which is the same point returned by our procedure) is almost optimal for F, i.e., $F(\bar{x}) \leq \mathrm{OPT}(F) + err(T, M', R)$. Since F and f are pointwise within $\pm 2\eta T$ of each other, the value of the solution $\bar{x}$ with respect to the original function f satisfies

$$\begin{aligned} f(\bar{x}) \leq F(\bar{x}) + 2\eta T &\leq \mathrm{OPT}(F) + err(T, M', R) + 2\eta T \\ &\leq \mathrm{OPT}(f) + err(T, M', R) + 4\eta T, \end{aligned}$$

which proves the desired result. □

3.1. Computing Lipschitz-extensible first-order pairs

In this section we describe the procedure that constructs the first-order pairs with a Lipschitz convex extension F that satisfies $\|F - f\|_\infty \leq 2\eta T$, proving Theorem 3.1. Before getting into the heart of the matter, we show that the latter

[2] We use *first-order pair* as just a more "visual" name for a pair in $\mathbb{R} \times \mathbb{R}^d$.

property can be significantly weakened: instead of requiring both $f(x) \geq F(x) - 2\eta T$ and $f(x) \leq F(x) + 2\eta T$ for all $x \in B(R)$, we can relax the latter to only hold for the queried points $x_1, \ldots, x_T$.

Lemma 3.2. Consider a sequence of points $x_1, \ldots, x_T \in B(R)$, and a sequence of first-order pairs $(\hat{f}_1, \hat{g}_1), \ldots, (\hat{f}_T, \hat{g}_T)$. Consider $\delta > 0$ and $M' \geq M$, and suppose that there is an M'-extension F of these first-order pairs that satisfies:

$$\underbrace{f(x) \geq F(x) - \delta}_{\text{approx. under-approximation}}\ , \qquad \forall x \in B(R) \tag{3.1}$$

$$\underbrace{f(x_t) \leq F(x_t) + \delta}_{\text{approx. queried values}}, \qquad \forall t = \{1, \ldots, T\}. \tag{3.2}$$

Then the first-order pairs have an M'-extension F' such that $\|F' - f\|_\infty \leq \delta$.

We sketch the main idea of the proof here and defer the complete proof to Appendix B.

Proof sketch: Since $F(x)$ only (approximately) under-approximates $f(x)$, other than at the queried points $F(x)$ may be much smaller than $f(x)$ for some x's. We fix this by taking the maximum of the function F and the function obtained by a "downward shift" of the original function f by $-\delta$ (which preserves Lipschitzness and convexity). The shift, together with property (3.2) guarantees that the function F is still "exposed" at the queried points x_t, and so it is still an extension of the first-order pairs $(\hat{f}_t, \hat{g}_t)$'s. □

Given Lemma 3.2, to prove Theorem 3.1 it suffices to do the following. Consider a sequence of points $x_1, \ldots, x_T \in B(R)$. Using an η-approximate first-order oracle to access the function f (at the points $x_1, \ldots, x_T$), we need to produce a sequence of first-order pairs $(\hat{f}_1, \hat{g}_1), \ldots, (\hat{f}_T, \hat{g}_T)$ in an online fashion that have an M-extension F achieving the approximations (3.1) and (3.2). We do this as follows.

At iteration t we maintain the function $F_t(x) := \max\{\hat{f}_\tau + \langle \hat{g}_\tau, x - x_\tau \rangle \ : \ \tau \leq t\}$, that is, the maximum of the linear functions induced by the first-order pairs $(\hat{f}_\tau, \hat{g}_\tau)$ constructed up to this point. We would like to define the pairs $(\hat{f}_\tau, \hat{g}_\tau)$ to guarantee that for all t, F_t is an M-extension for these pairs, and satisfies (3.1) and (3.2) for $x_1, \ldots, x_t$. In this case, $F = F_T$ gives the desired function.

For that, suppose the above holds for $t - 1$; we will show how to define $(\hat{f}_t, \hat{g}_t)$ to maintain this invariant for t. We should think of constructing F_t by taking the maximum of F_{t-1} and a new linear function $\hat{f}_t + \langle \hat{g}_t, x_\tau - x_t \rangle$. To ensure that F_t is an extension of the first-order pairs thus far, we need to make sure that:

1. $\hat{f}_t \geq F_{t-1}(x_t)$. This is necessary to ensure that $F_t(x_t) = \hat{f}_t$, and also guarantees $\hat{g}_t \in \partial F_t(x_t)$.

2. $\hat{f}_t + \langle \hat{g}_t, x_\tau - x_t \rangle \leq F_{t-1}(x_\tau), \quad \forall \tau \leq t - 1$. This is necessary to ensure that $F_t(x_\tau) = F_{t-1}(x_\tau) = \hat{f}_\tau$, and also guarantees $\partial F_t(x_\tau) \supseteq \partial F_{t-1}(x_\tau) \ni \hat{g}_\tau$.

To construct $(\hat{f}_t, \hat{g}_t)$ with these properties, we probe the approximate first-order oracle for f at x_t, and receive an answer $(\tilde{f}_t, \tilde{g}_t)$. If setting $(\hat{f}_t, \hat{g}_t) = (\tilde{f}_t, \tilde{g}_t)$ violates the first item above, we simply use the first-order information of F_{t-1} at x_t, i.e., we set $\hat{f}_t = F_{t-1}(x_t)$ and $\hat{g}_t \in \partial F_{t-1}(x_t)$.

If the second item above is violated instead, we shift the value $\tilde{f}_t$ down as little as possible to ensure the desired property, i.e., we set $(\hat{f}_t, \hat{g}_t) = (\tilde{f}_t - s^*, \tilde{g}_t)$ for appropriate $s^* > 0$. With this shifted value, the first item may now be violated, in which case we again just use the current first-order information of F_{t-1}.

These steps are formalized in the following procedure.

Procedure 2.

```
For each $t = 1, \ldots, T$:
  1. Query the $\eta$-approximate oracle for $f$ at $x_t$, receiving the first-order pair $(\tilde{f}_t, \tilde{g}_t)$
  2. Let $s^* := \min\{s \geq 0 : \tilde{f}_t - s + \langle \tilde{g}_t, x_\tau - x_t \rangle \leq F_{t-1}(x_\tau), \forall \tau \leq t - 1\}$
  3. If $\tilde{f}_t - s^* \geq F_{t-1}(x_t)$, then set $(\hat{f}_t, \hat{g}_t) = (\tilde{f}_t - s^*, \tilde{g}_t)$
  4. Else, set $\hat{f}_t = F_{t-1}(x_t)$ and let $\hat{g}_t$ be subgradient in $\partial F_{t-1}(x_t)$
  5. Set $F_t(x) = \max\{F_{t-1}(x)\ ,\ \hat{f}_t + \langle \hat{g}_t, x - x_t \rangle\}$.
```

We remark that this requires storing historical values of $\tilde{f}_t$ and $\tilde{g}_t$ (this seems unavoidable to ensure convexity of F_t). We now prove that the functions F_t's have the desired properties.

Lemma 3.3. For every $t = 1, \ldots, T$, the function F_t is an $(M + \frac{\eta}{2R})$-extension of the first-order information pairs $(\hat{f}_1, \hat{g}_1), \ldots, (\hat{f}_t, \hat{g}_t)$.

Proof. Since F_t is the maximum over affine functions, it is convex. Moreover, all of its subgradients come from the set $\{\tilde{g}_\tau\}_\tau$, and by the approximation guarantee of the oracle we have that for some subgradient $g \in \partial f(x_\tau)$, $\|\tilde{g}\|_2 \leq \|\tilde{g} - g\|_2 + \|g\|_2 \leq \frac{\eta}{2R} + M$, where we used that fact that f is M-Lipschitz; thus, F_t is $(M + \frac{\eta}{2R})$-Lipschitz.

We prove by induction on t that F_t is an extension of the desired pairs (the base case $t = 1$ can be readily verified). Recall $F_t(x_t) = \max\{F_{t-1}(x), H(x)\}$, where $H(x) := \hat{f}_t + \langle \hat{g}, x - x_t \rangle$. By the definition of s^*, for all $x = x_1, \dots, x_{t-1}$, this maximum is achieved by the function F_{t-1}, giving, by induction, that for all $\tau \le t-1$, $F_t(x_\tau) = F_{t-1}(x_\tau) = \hat{f}_\tau$; this also implies that for such τ's, $\partial F_t(x_\tau) \supseteq \partial F_{t-1}(x_\tau) \ni \hat{g}_\tau$, the last inclusion again following by induction. These give the extension property for the pairs $(\hat{f}_\tau, \hat{g}_\tau)$ with $\tau \le t-1$.

It remains to verify that this also holds for $\tau = t$. Now the maximum in the definition of $F_t(x_t)$ is achieved by the function H: if $\tilde{f}_t - s^* \ge F_{t-1}(x_t)$, the procedure sets $\hat{f}_t = \tilde{f}_t - s^*$ and we have $H(x_t) = \hat{f}_t \ge F_{t-1}(x_t)$; otherwise the procedure sets $\hat{f}_t = F_{t-1}(x_t)$ and so $H(x_t) = F_{t-1}(x_t)$. Again this implies that $\partial F_t(x_t) \supseteq \partial H(x_t) = \{\hat{g}_t\}$. This concludes the proof of the lemma. □

Finally, we show that the functions F_t approximate f according to (3.1) and (3.2).

Lemma 3.4. For every $t = 1, \dots, T$, the F_t satisfies inequalities (3.1) and (3.2) with $\delta = 2\eta t$.

Proof. Again we prove this by induction on t. Fix t. Let $\Delta := \tilde{f}_t - f(x_t)$ be the error the inexact oracle makes on the function value. We claim that the shift s^* used in iteration t of Procedure 2 satisfies $s^* \le \max\{0, \Delta + 2\eta t\}$. To see this, the η-approximation of the oracle guarantees that there is a subgradient $g \in \partial f(x_t)$ such that $\|\tilde{g}_t - g\|_2 \le \frac{\eta}{2R}$, and so for every $\tau \le t-1$

$$\begin{aligned} &\tilde{f}_t + \langle \tilde{g}_t, x_\tau - x_t \rangle \\ &= \Delta + \underbrace{f(x_t) + \langle g, x_\tau - x_t \rangle}_{\le f(x_\tau)} + \underbrace{\langle \tilde{g}_t - g, x_\tau - x_t \rangle}_{\le \|\tilde{g}_t - g\|_2 \|x_\tau - x_t\|_2 \le \eta} \\ &\le F_{t-1}(x_\tau) + \Delta + 2t\eta, \end{aligned} \tag{3.3}$$

the first underbrace following since g is a subgradient of f, and the last inequality following from the induction hypothesis $F_{t-1}(x_\tau) \ge f(x_\tau) - 2(t-1)\eta$ (inequality (3.2)); the optimality of s^* then guarantees that it is at most $\max\{0, \Delta + 2\eta t\}$, proving the claim.

Now we show that F_t satisfies the desired bounds, namely $F_t(x_\tau) \ge f(x_\tau) - 2\eta t$ for all $\tau \le t$, and $F_t(x) \le f(x) + 2\eta t$ for all $x \in B(R)$. From the inductive hypothesis, for $\tau \le t-1$ we have $F_t(x_\tau) \ge F_{t-1}(x_\tau) \ge f(x_\tau) - 2\eta(t-1)$, giving the first bound for these x_τ. For x_t, notice that $F_t(x_t) \ge \tilde{f}_t - s^*$: if Line 3 of the procedure is executed we have $F_t(x_t) \ge \hat{f}_t = \tilde{f}_t - s^*$, and if Line 4 is executed then it means $\tilde{f}_t - s^* < F_{t-1}(x_t)$ and so $F_t(x_t) = F_{t-1}(x_t) > \tilde{f}_t - s^*$. Then

$$\begin{aligned} F_t(x_t) &\ge \tilde{f}_t - s^* \ge \tilde{f}_t - \max\{0, \Delta - 2\eta t\} \\ &\ge \max\{f(x_t) - \eta\,,\, f(x_t) - 2\eta t\} = f(x_t) - 2\eta t, \end{aligned}$$

where in the second inequality we used the upper bound on the shift $s^* \le \max\{0, \Delta + 2\eta t\}$, and in the next inequality we used the guarantee $|\tilde{f}_t - f(x_t)| \le \eta$ from the approximate oracle.

For the upper bound $F_t(x) \le f(x) + 2\eta t$, when Line 3 of the procedure is executed we have $F_t(x) = F_{t-1}(x) \le f(x) + 2\eta(t-1)$, the last inequality from the inductive hypothesis. When Line 4 is executed instead, we have $\hat{f}_t = \tilde{f}_t - s^*$, and the same development as in (3.3) reveals that

$$\begin{aligned} \hat{f}_t + \langle \tilde{g}_t, x - x_t \rangle \le& \tilde{f}_t + \langle \tilde{g}_t, x - x_t \rangle \\ &\le f(x) + \Delta + \eta \le f(x) + 2\eta, \end{aligned}$$

where the last inequality again uses that $\Delta = \tilde{f}_t - f(x_t) \le \eta$ due to the guarantee of the approximate oracle. Thus, $F_t(x) \le \max\{f(x) + 2\eta(t-1), f(x) + 2\eta\} \le f(x) + 2\eta t$, giving the desired bound. This concludes the proof of the lemma. □

Combining Lemmas 3.2, 3.3, and 3.4 shows that the first-order pairs produced by Procedure 2 satisfies the properties stated in Theorem 3.1, finally concluding its proof.

4. Universal transfer for smooth functions

In this section we prove our transfer theorem for smooth functions stated in Theorem 2.3. Recall that a function f is α-smooth if it has α-Lipschitz gradients:

$$\forall x, y \in \mathbb{R}^d, \quad \|\nabla f(x) - \nabla f(y)\| \le \alpha \|x - y\|.$$

As in the proof of the previous transfer theorem, the core element is the following: Given the sequence of iterates $x_1, \dots, x_t$ of a black-box optimization algorithm and access to an approximate first-order oracle to the smooth objective function f, construct in an online fashion first-order pairs $(\hat{f}_t, \hat{g}_t)$ and, implicitly, a *smooth* function S close to the original f such that $(\hat{f}_t, \hat{g}_t)$ provide exactly the value and gradient of S at x_t.

Theorem 4.1 (Online first-order smooth-extensibility)**.** Consider an α-smooth, M-Lipschitz convex function $f : B(R) \to \mathbb{R}$, and a sequence of points $x_1, \dots, x_T \in B(R)$. Then there is an online procedure that given η-approximate first-order oracle access to f, produces first-order pairs $(\hat{f}_1, \hat{g}_1), \dots, (\hat{f}_T, \hat{g}_T)$ that have an α'-smooth $(M + \frac{\eta}{2R})$-extension $S : B(R) \to \mathbb{R}$ satisfying $\|S - f\|_\infty \le 5\eta\,(T+2)$, where $\alpha' = \alpha\,\sqrt{d}\,\Big(4\sqrt{5 \cdot (T+1)} + 3\Big)$. Moreover, the procedure only probes the approximate oracle at the given points $x_1, \dots, x_T$.

In the previous section, the extension was created by adding a new linear function at every iteration; this produced the piecewise linear (non-smooth) functions F_t in the previous section. Having to construct a *smooth* extension creates a challenge. Our approach is to apply a *smoothing* procedure to these piecewise linear functions, in an online manner.

One issue is that most standard smoothing procedures (e.g., via inf-convolution (Beck & Teboulle, 2012) or Gaussian smoothing (Nesterov, 2005)) may use the values of the non-smooth base function over the whole domain; in our online construction, at a given point in time we have determined the value of the function only in a neighborhood of the previous iterates, and the updated functions can change at points outside these small neighborhoods. Thus, we employ a localized smoothing procedure. Moreover, we need the procedure to leverage the fact that the non-smooth base function is close to a smooth one, and produce stronger smoothing guarantees by making use thereof. We start by describing this smoothing technique and its properties, and then describe the full procedure that gives Theorem 4.1.

Randomized smoothing of almost smooth functions. Given a function $h : \mathbb{R}^d \to \mathbb{R}$ and a radius $r > 0$, we define the smoothed function h_r by $h_r(x) := \mathbb{E}h(x + rU)$, where U is uniformly distributed on the unit ball $B(0, 1)$. It is well-known that when h is convex and M-Lipschitz, then h_r is differentiable, also M-Lipschitz, and, most importantly, is $\frac{M\sqrt{d}}{r}$-smooth (Yousefian et al., 2012). However, we show that the smoothing parameter can be significantly improved when the function h is already close to a smooth function. The proof is deferred to Appendix A.1.

Lemma 4.2. Let $h : \mathbb{R}^d \to \mathbb{R}$ be a convex function such that there exists an α-smooth convex function f with $\|h - f\|_\infty \leq \varepsilon$. Then the smoothed function h_r satisfies:

1. h_r is $\left(\frac{4\sqrt{\alpha\varepsilon d}}{r} + 3\alpha\sqrt{d}\right)$-smooth
2. $\|h_r - f\|_\infty \leq \varepsilon + \frac{\alpha r^2}{2}$.

Construction of the smooth-extension. As mentioned, in each iteration t we will maintain a piecewise linear function F_t constructed very similarly to the proof of Theorem 3.1. Now we will also maintain the smoothened version $(F_t)_r$ of this function that uses the randomized smoothing discussed above (for a particular value of r). Our transfer procedure then returns the first-order information $\hat{f}_t := (F_t)_r(x_t)$ and $\hat{g}_t := \nabla(F_t)_r(x_t)$ of the latter. The final smooth function $S : B(R) \to \mathbb{R}$ compatible with the first-order information returned by the procedure will be given, as in Lemma 3.2, by taking the maximum between the final F_T and a shifted version of the original function f.

The main difference in how the functions F_t's are constructed, compared to the proof of Theorem 3.1, is the following. Previously, in order to ensure that F_T (and so the final extension) was compatible with the first-order pairs output in earlier iterations, we needed to "protect" the points x_t and ensure that the function values and gradients at these points did not change over time, e.g., we needed $F_T(x_t) = F_t(x_t)$. But now the first-order pair output for the query point x_t depends not only on the value of F_t at x_t, but also on the values on the whole *ball* $B(x_t, r)$ that are used to determine the smoothed function $(F_t)_r$ at x_t. Thus, we will now need to "protect" these balls and ensure that the function values over them do not change in later iterations.

We now formalize the construction of the functions F_t, the first-order information returned, and the final extension S in Procedure 3.

Procedure 3.

Set $r = \sqrt{\eta/\alpha}$. For each $t = 1, \ldots, T$:

1. Query the η-approximate oracle for f at x_t, receiving the first-order pair $(\tilde{f}_t, \tilde{g}_t)$
2. Define the function F_t by setting $F_t(x) = \max\{F_{t-1}(x)\, ,\, \tilde{f}_t + \langle \tilde{g}_t, x - x_t\rangle - (4\eta t + \alpha r^2 t + 2\eta)\}$ for all x
3. Output the first-order information of the randomly smoothed function $(F_t)_r$: $\hat{f}_t := (F_t)_r(x_t)$ and $\hat{g}_t := \nabla(F_t)_r(x_t)$

Define the function $S : B(R) \to \mathbb{R}$ by $S = (\max\{F_T, f - 4\eta(T + 1) + \alpha r^2(T + 1)\})_r$, where max denotes pointwise maximum.

The proof that this procedure indeed yields Theorem 4.1 is presented in Appendix A.2.

Impact Statement

This paper presents work whose goal is to advance the fields of Machine Learning and Optimization. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Bandi, H., Bertsimas, D., and Mazumder, R. Learning a mixture of gaussians via mixed-integer optimization. *INFORMS Journal on Optimization*, 1(3):221–240, 2019.

Basu, A. Complexity of optimizing over the integers. *Mathematical Programming, Series B*, 200:739–780, 2023.

Basu, A., Jiang, H., Kerger, P., and Molinaro, M. Information complexity of mixed-integer convex optimization. *arXiv preprint arXiv:2308.11153*, 2023.

Beck, A. and Teboulle, M. Smoothing and first order methods: A unified framework. *SIAM Journal on Optimization*, 22(2): 557–580, 2012. doi: 10.1137/100818327. URL https://doi.org/10.1137/100818327.

Bertsekas, D. P. Stochastic optimization problems with nondifferentiable cost functionals. *Journal of Optimization Theory and Applications*, 12:218–231, 1973.

Bertsimas, D., King, A., and Mazumder, R. Best subset selection via a modern optimization lens. *The Annals of Statistics*, 44(2): 813–852, 2016.

Bubeck, S. Convex optimization: Algorithms and complexity. *Found. Trends Mach. Learn.*, 8(3–4):231–357, nov 2015. ISSN 1935-8237. doi: 10.1561/2200000050. URL https://doi.org/10.1561/2200000050.

Cottle, R. W., Pang, J.-S., and Stone, R. E. *The linear complementarity problem*, volume 60. Siam, 2009.

d'Aspremont, A. Smooth optimization with approximate gradient. *SIAM Journal on Optimization*, 19(3):1171–1183, 2008.

Dedieu, A., Hazimeh, H., and Mazumder, R. Learning sparse classifiers: Continuous and mixed integer optimization perspectives. *The Journal of Machine Learning Research*, 22(1):6008–6054, 2021.

Devolder, O., Glineur, F., and Nesterov, Y. First-order methods of smooth convex optimization with inexact oracle. *Mathematical Programming*, 146:37–75, 2014.

Dey, S. S., Mazumder, R., and Wang, G. Using ℓ_1-relaxation and integer programming to obtain dual bounds for sparse pca. *Operations Research*, 70(3):1914–1932, 2022.

d'Aspremont, A., Scieur, D., and Taylor, A. Acceleration methods. *Foundations and Trends® in Optimization*, 5(1-2):1–245, 2021. ISSN 2167-3888. doi: 10.1561/2400000036. URL http://dx.doi.org/10.1561/2400000036.

Hazimeh, H., Mazumder, R., and Saab, A. Sparse regression at scale: Branch-and-bound rooted in first-order optimization. *Mathematical Programming*, 196(1-2):347–388, 2022.

Hazimeh, H., Mazumder, R., and Radchenko, P. Grouped variable selection with discrete optimization: Computational and statistical perspectives. *The Annals of Statistics*, 51(1):1–32, 2023.

Hintermüller, M. A proximal bundle method based on approximate subgradients. *Computational Optimization and Applications*, 20:245–266, 2001.

Kiwiel, K. C. A proximal bundle method with approximate subgradient linearizations. *SIAM Journal on optimization*, 16(4):1007–1023, 2006.

Lan, G. *Convex optimization under inexact first-order information*. Georgia Institute of Technology, 2009.

Mazumder, R. and Radchenko, P. The discrete dantzig selector: Estimating sparse linear models via mixed integer linear optimization. *IEEE Transactions on Information Theory*, 63(5):3053–3075, 2017.

Nemirovski, A. Efficient methods in convex programming. *Lecture notes*, 1994.

Nesterov, Y. A method of solving a convex programming problem with convergence rate $o(\frac{1}{k^2})$. *Dokl. Akad. Nauk SSSR*, 269(3):543–547, 1983.

Nesterov, Y. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103:127–152, 2005.

Nesterov, Y. E. *Introductory Lectures on Convex Optimization*, volume 87 of *Applied Optimization*. Kluwer Academic Publishers, Boston, 2004. ISBN 1-4020-7553-7.

Nesterov, Y. E. *Lectures on convex optimization*, volume 137. Springer, 2018.

Polyak, B. Introduction to optimization. *Translations Series in Mathematics and Engineering. New York: Optimization Software Inc. Publications Division*, 1987.

Schmidt, M., Roux, N., and Bach, F. Convergence rates of inexact proximal-gradient methods for convex optimization. *Advances in neural information processing systems*, 24, 2011.

Shor, N. Z. *Minimization methods for non-differentiable functions*. Springer Series in Computational Mathematics, 1985.

Wang, J.-K. and Abernethy, J. D. Acceleration through optimistic no-regret dynamics. *Advances in Neural Information Processing Systems*, 31, 2018.

Yousefian, F., Nedić, A., and Shanbhag, U. V. On stochastic gradient and subgradient methods with adaptive steplength sequences. *Automatica*, 48(1):56–67, 2012. ISSN 0005-1098. doi: https://doi.org/10.1016/j.automatica.2011.09.043. URL https://www.sciencedirect.com/science/article/pii/S0005109811004833.

Appendix

A. Universal transfer for smooth functions

In this section we present the missing proofs for our transfer theorem for smooth functions from Section 4. We start by recalling the definition of a smooth function.

Definition A.1. A function $f : \mathbb{R}^d \to \mathbb{R}$ is said to be α-smooth if it is differentiable and its gradient is Lipschitz continuous with a Lipschitz constant α, namely

$$\forall x, y \in \mathbb{R}^d, \quad \|\nabla f(x) - \nabla f(y)\| \le \alpha \|x - y\|.$$

An α-smooth function possesses the following useful upper bounding property: for $x, y \in \mathbb{R}^n$:

$$f(y) \le f(x) + \langle \nabla f(x), y - x \rangle + \frac{\alpha}{2}\|y - x\|^2. \tag{A.1}$$

A.1. Proof of Lemma 4.2

Let h and f be the convex functions satisfying the statement of the lemma, i.e., $\|h - f\|_\infty \le \varepsilon$ and f is α-smooth. Recall that the smoothed function h_r is defined as $h_r(x) = \mathbb{E}h(x + rU)$, where U is a random vector uniformly distributed on the unit ball $B(0, 1)$.

The first observation is that since h is close to f and the latter is smooth, their (sub)gradients are close to each other; the same also holds between h_r and f.

Lemma A.2. For every x:

1. $\|\partial h(x) - \nabla f(x)\| \le 2\sqrt{\alpha\varepsilon}$ for every subgradient $\partial h(x)$

2. $\|\nabla h_r(x) - \nabla f(x)\| \le 2\sqrt{\alpha\varepsilon} + \alpha r$.

Proof. To prove the first item, consider any point $y \in \mathbb{R}^d$. Using α-smoothness of f, $\|h - f\|_\infty \le \varepsilon$, and convexity of h, we have

$$f(x) + \langle \nabla f(x), y - x\rangle + \frac{\alpha}{2}\|x-y\|^2 \;\ge\; f(y) \;\ge\; h(y) - \varepsilon \\ \ge\; h(x) + \langle \nabla h(x), y - x\rangle - \varepsilon,$$

and so

$$\langle \nabla h(x) - \nabla f(x), y - x\rangle \le f(x) - h(x) + \varepsilon + \frac{\alpha}{2}\|x-y\|^2 \\ \le 2\varepsilon + \frac{\alpha}{2}\|x-y\|^2 .$$

Then setting y so that $y - x = \frac{1}{\alpha}(\nabla h(x) - \nabla f(x))$ gives

$$\frac{1}{2\alpha}\|\nabla h(x) - \nabla f(x)\|^2 \le 2\varepsilon,$$

which gives the first item of the lemma.

For the second item, again let U be uniformly distributed in $B(0,1)$. This random variable is sufficiently regular that gradients and expectations commute, namely $\nabla h_r(x) = \nabla(\mathbb{E}\, h(x + rU)) = \mathbb{E}\,\partial h(x+rU)$, were ∂h denotes any subgradient of h (Bertsekas, 1973).Then applying Jensen's inequality we get

$$\|\nabla h_r(x) - \nabla f(x)\| = \|\mathbb{E}\,\partial h(x+rU) - \nabla f(x)\| \\ \le \mathbb{E}\,\|\partial h(x+rU) - \nabla f(x)\|.$$

Also, for any unit-norm vector u we have

$$\|\partial h(x+ru) - \nabla f(x)\| \le \|\partial h(x+ru) - \nabla f(x+ru)\| \\ + \|\nabla f(x+ru) - \nabla f(x)\| \\ \le 2\sqrt{\alpha\varepsilon} + \alpha r,$$

where the last inequality follows from Item 1 of the lemma and α-smoothness of f (which is equivalent to $\|\nabla f(z) - \nabla f(z')\| \le \alpha\|z - z'\|$ (Nesterov, 2018)). This concludes the proof. □

The second element that we will need is a bound on the total variation between the the uniform distributions on the two same-radius balls with different centers (see for example Lemma 8 of (Yousefian et al., 2012) for a proof). We state it more conveniently in the language of coupling.

Lemma A.3. Let $X \in \mathbb{R}^d$ be the uniformly distributed on $B(x,r)$ and $Y \in \mathbb{R}^d$ be uniformly distributed on $B(y,r)$. Then there is a random variable $(X', Y') \in \mathbb{R}^{2d}$ where X' has the same distribution as X and Y' the same distribution as Y, and where $\Pr(X' \neq Y') \le \frac{\|x-y\|\sqrt{d}}{r}$.

We are now ready to prove Lemma 4.2.

Proof of Lemma 4.2. Item 1: We prove that $\|\nabla h_r(x) - \nabla h_r(y)\| \le (\frac{4\sqrt{\alpha\varepsilon d}}{r} + 3\alpha\sqrt{d})\|x - y\|$ for all x, y. In fact, it suffices to prove this for x, y where $\|x - y\| \le r$, since the inequality can then be chained to obtain the result for any pair of points.

Then fix x, y with $\|x - y\| \le r$. Using the notation from Lemma A.3, $\nabla h_r(x) = \mathbb{E}\,\partial h(X')$ and $\nabla h_r(y) = \mathbb{E}\,\partial h(Y')$ and $\Pr(X' \neq Y') \le \frac{\|x-y\|\sqrt{d}}{r}$. Applying Jensen's inequality,

$$\|\nabla h_r(x) - \nabla h_r(y)\| \le \mathbb{E}_{X',Y'}\|\partial h(X') - \partial h(Y')\| \\ \le \frac{\|x-y\|\sqrt{d}}{r} \cdot \max_{x',y' \in B(x,r)\cup B(y,r)} \|\partial h(x') - \partial h(y')\|.$$

We upper bound the last term by applying triangle inequality and then Lemma A.2:

$$\|\partial h(x') - \nabla h(y')\| \le \|\partial h(x') - \nabla f(x')\| \\ + \|\nabla f(x') - \nabla f(y')\| \\ + \|\nabla f(y') - \partial h(y')\| \\ \le 4\sqrt{\alpha\varepsilon} + \alpha\,\|x' - y'\| \\ \le 4\sqrt{\alpha\varepsilon} + 3\alpha r,$$

where the second inequality uses that f is α-smooth, and the last inequality uses the assumption $\|x - y\| \le r$. Plugging this into the previous inequality gives

$$\|\nabla h_r(x) - \nabla h_r(y)\| \le \left(\frac{4\sqrt{\alpha\varepsilon d}}{r} + 3\alpha\sqrt{d}\right) \cdot \|x - y\|,$$

as desired.

Second item: We now show that $\|h_r - f\|_\infty \le \varepsilon + \frac{\alpha r^2}{2}$. Fix $x \in \mathbb{R}^d$, and again let U be uniformly distributed in the unit ball. Using the assumption $\|h - f\|_\infty \le \varepsilon$ and convexity of f, we have

$$h(x+rU) \ge f(x+rU) - \varepsilon \ge f(x) + \langle \nabla f(x), rU\rangle - \varepsilon.$$

Since U has mean zero, taking expectations gives $h_r(x) \ge f(x) - \varepsilon$. Similarly, since f is α-smooth

$$h(x+rU) \le f(x+rU) + \varepsilon \\ \le f(x) + \langle \nabla f(x), rU\rangle + \frac{\alpha}{2}\|rU\|^2 + \varepsilon,$$

and taking expectations gives $h_r(x) \le f(x) + \frac{\alpha r^2}{2} + \varepsilon$. Together, these yield $|h_r(x) - f(x)| \le \varepsilon + \frac{\alpha r^2}{2}$, thus proving the result. This concludes the proof of the theorem. □

A.2. Proof of Theorem 4.1

Throughout this section, fix an α-smooth M-Lipschitz function $f : B(R) \to \mathbb{R}$. Recall that we have a sequence of queried points $x_1, \ldots, x_T \in B(R)$ and access to an η-approximate first-order oracle for f. Our goal is to produce, in an online fashion, a sequence of first-order pairs $(\hat{f}_1, \hat{g}_1), \ldots, (\hat{f}_T, \hat{g}_T)$ for the queried points and a function S that is smooth, Lipschitz, and compatible with these first-order pairs (i.e., $S(x_t) = \hat{f}_t$ and $\nabla S(x_t) = \hat{g}_t$).

As mentioned, in each iteration t we will keep a piecewise linear function F_t and their smoothened version $(F_t)_r$ (by using the randomized smoothing from the previous section for a specific value of r). Our transfer procedure then returns the first-order information $\hat{f}_t := (F_t)_r(x_t)$ and $\hat{g}_t := \nabla(F_t)_r(x_t)$ of the latter. The final smooth function $S : B(R) \to \mathbb{R}$ compatible with the first-order information output by the procedure will be given, as in Lemma 3.2, by using the maximum between the final F_T and a shifted version of the original function f. Also recall that in order to ensure the compatibility of S with the first-order information $(\hat{f}_t, \hat{g}_t)$ output throughout the process, we need to "protect" the points x_t and ensure that the function values and gradients at these points did not change across iterations, i.e. $(F_T)_r(x_t) = (F_t)_r(x_t)$ and $\nabla(F_T)_r(x_t) = \nabla(F_t)_r(x_t)$. Since $(F_{t'})_r(x_t)$ depends on the values of $F_{t'}$ at the ball $B(x_t, r)$ around x_t, we need to "protect" $F_{t'}$ on these balls, namely to have $F_T(x) = F_t(x)$ for all $x \in B(x_t, r)$.

For convenience, we recall the exact construction of the functions F_t, the first-order information returned, and the final extension S. In hindsight, set $r := \sqrt{\eta/\alpha}$, and for every t define the shift $s_t := 4\eta t + \alpha r^2 t$.

Procedure 3.

For each $t = 1, \ldots, T$:

1. Query the η-approximate oracle for f at x_t, receiving the first-order pair $(\tilde{f}_t, \tilde{g}_t)$
2. Define the function F_t by setting $F_t(x) = \max\{F_{t-1}(x)\,,\, \tilde{f}_t + \langle \tilde{g}_t, x - x_t\rangle - (s_t + 2\eta)\}$ for all x
3. Output the first-order information of the randomly smoothed function $(F_t)_r$: $\hat{f}_t := (F_t)_r(x_t)$ and $\hat{g}_t := \nabla (F_t)_r(x_t)$

Define the function $S : B(R) \to \mathbb{R}$ by $S = (\max\{F_T, f - s_{T+1}\})_r$, where $\max$ denotes pointwise maximum.

We now prove the main properties of the functions F_t, formulated in the following lemma. The first two are similar to (3.1) and (3.2) used in our non-smooth transfer result and guarantee, loosely speaking, that F_t is close to the original function f. The third property is precisely the "ball protection" idea discussed above.

Lemma A.4. For all t, the function F_t satisfies the following:

1. $F_t(x) \leq f(x)$ for every $x \in B(R)$
2. For every $t' \leq t$, we have $F_t(x) \geq f(x) - s_{t+1}$ for all $x \in B(x_{t'}, \sqrt{2}\, r)$
3. For every $t' \leq t$, we have $F_t(x) = F_{t'}(x)$ for every $x \in B(x_{t'}, \sqrt{2}\, r)$. In particular $(F_t)_r(x_{t'}) = (F_{t'})_r(x_{t'})$ and $\nabla (F_t)_r(x_{t'}) = \nabla (F_{t'})_r(x_{t'})$.

Proof. We prove these properties by induction on t.

First item: Since the property holds by induction for F_{t-1} and $F_t(x) = \max\{F_{t-1}(x)\,,\, \tilde{f}_t + \langle \tilde{g}_t, x - x_t\rangle - (s_t + 2\eta)\}$, it suffices to show that

$$\tilde{f}_t + \langle \tilde{g}_t, x - x_t\rangle - (s_t + 2\eta) \leq f(x) \tag{A.2}$$

for all $x \in B(R)$. For that, since $(\tilde{f}_t, \tilde{g}_t)$ comes from an η-approximate first-order oracle, by definition $|\tilde{f}_t - f(x_t)| \leq \eta$ and $\|\tilde{g}_t - \nabla f(x_t)\| \leq \frac{\eta}{2R}$; in particular, $|\langle \tilde{g}_t - \nabla f(x_t), x - x_t\rangle| \leq \|\tilde{g}_t - \nabla f(x_t)\| \|x - x_t\| \leq \eta$ for every $x \in B(R)$ (since also $x_t \in B(R)$, by assumption). Then using convexity of f we get

$$\begin{aligned} f(x) &\geq f(x_t) + \langle \nabla f(x_t), x - x_t\rangle \\ &\geq \tilde{f}_t + \langle \tilde{g}_t, x - x_t\rangle - 2\eta, \end{aligned} \tag{A.3}$$

which implies (A.2) as desired.

Second item: Again since this property holds by induction for F_{t-1}, it suffices to show

$$\tilde{f}_t + \langle \tilde{g}_t, x - x_t\rangle - (s_t + 2\eta) \geq f(x) - s_{t+1} \tag{A.4}$$

for all $x \in B(x_t, \sqrt{2}\, r)$. Since f is α-smooth, for every such x we have

$$\begin{aligned} f(x) &\leq f(x_t) + \langle \nabla f(x_t), x - x_t\rangle + \frac{\alpha}{2}\|x - x_t\|^2 \\ &\leq \tilde{f}_t + \langle \tilde{g}_t, x - x_t\rangle + 2\eta + \alpha r^2. \end{aligned} \tag{A.5}$$

Since $s_{t+1} = s_t + 4\eta + \alpha r^2$, reorganizing the terms gives (A.4) as desired.

Third item: To show that for every $t' \leq t$, we have $F_t(x) = F_{t'}(x)$ for every $x \in B(x_{t'}, \sqrt{2}\, r)$, it suffices to show that for every $t' < t$

$$\tilde{f}_t + \langle \tilde{g}_t, x - x_t\rangle - (s_t + 2\eta) \leq F_{t-1}(x) \tag{A.6}$$

for all $x \in B(x_{t'}, \sqrt{2}\, r)$. For that, first notice that for all $t' < t$ we have $F_{t-1} \geq F_{t'}$, and the latter can be lower bounded by the affine term added during iteration t'. Combining this with (A.5), applied to iteration t', we get for all $x \in B(x_{t'}, \sqrt{2}\, r)$

$$\begin{aligned} F_{t-1}(x) &\geq \tilde{f}_{t'} + \langle \tilde{g}_{t'}, x - x_{t'}\rangle - (s_{t'} + 2\eta) \\ &\geq f(x) - (s_{t'} + 4\eta + \alpha r^2) \\ &\geq \tilde{f}_t + \langle \tilde{g}_t, x - x_t\rangle - (s_{t'} + 6\eta + \alpha r^2), \end{aligned}$$

where the last inequality uses (A.3). Since $s_t \geq s_{t'} + 4\eta + \alpha r^2$, this implies (A.6) as desired.

To conclude the proof of this item, notice that $(F_t)_r(x_{t'})$ (respectively $(F_{t'})_r(x_{t'})$) only depends on the values of F_t (resp. $F_{t'}$) on the ball $B(x_{t'}, r)$. Since we just showed the value of F_t and $F_{t'}$ agree on this ball, we get $(F_t)_r(x_{t'}) = (F_{t'})_r(x_{t'})$. Similarly, the gradient $\nabla (F_t)_r(x_{t'})$ only depends on the values of F_t on an arbitrarily small open neighborhood of the ball $B(x_{t'}, r)$, and the same holds for $\nabla (F_{t'})_r(x_{t'})$. Since the bigger ball $B(x_{t'}, \sqrt{2}\, r)$ contains such a neighborhood, we again obtain $\nabla (F_t)_r(x_{t'}) = \nabla (F_{t'})_r(x_{t'})$. This concludes the proof of the lemma. □

We are now ready to prove Theorem 4.1.

Proof of Theorem 4.1. We need to prove that the function $S : B(R) \to \mathbb{R}$ defined in Procedure 3 satisfies:

1. $\|S - f\|_\infty \leq s_{T+1} + \frac{\alpha r^2}{2}$.
2. S is $\left(\frac{4\sqrt{\alpha \cdot d \cdot s_{T+1}}}{r} + 3\alpha\sqrt{d}\right)$-smooth
3. S is $(M + \frac{\eta}{2R})$-Lipschitz
4. S is an extension for the first-order pairs $(\hat{f}_t, \hat{g}_t)$ output by the procedure

First item: Define the function $\bar{S} := \max\{F_t(x), f(x) - s_{T+1}\}$, so $S = \bar{S}_r$. Using Item 1 of Lemma A.4, we see that $\bar{S}(x) \leq f(x)$ for all $x \in B(R)$, and by definition we have $\bar{S}(x) \geq f(x) - s_{T+1}$, thus $\|\bar{S} - f\|_\infty \leq s_{T+1}$. Then using Item 2 of Lemma 4.2 we get $\|S - f\|_\infty \leq s_{T+1} + \frac{\alpha r^2}{2}$.

Second item: This follows directly from the above fact that $\|\bar{S} - f\|_\infty \leq s_{T+1}$ and the guarantees for randomized smoothing given in Lemma 4.2.

Third item: The subgradients of F_T are (a convex combination of a subset of the) vectors $\tilde{g}_t$, and so F_T is $(\max_t \|\tilde{g}_t\|)$-Lipschitz. Since the vectors came from an η-approximate oracle for f, we have $\|\tilde{g}_t - \nabla f(x_t)\| \leq \frac{\eta}{2R}$, and since f is M-Lipschitz we get $\|\tilde{g}_t\| \leq M + \frac{\eta}{2R}$; it follows that F_T is $(M + \frac{\eta}{2R})$-Lipschitz. Next, the subgradients of $\bar{S}$ come either from subgradients of F_T

or gradients of f (or a convex combination thereof), and so $\bar{S}$ is $\max\{M + \frac{\eta}{2R}, M\} = M + \frac{\eta}{2R}$ Lipschitz. Finally, for every x we have (U being uniformly distributed in the unit ball again)

$$\|\nabla S(x)\| = \|\mathbb{E}\partial \bar{S}(x + rU)\| \leq \mathbb{E}\|\partial \bar{S}(x + rU)\| \leq M + \frac{\eta}{2R},$$

where $\partial \bar{S}(x + rU)$ denotes any subgradient at $x + rU$ and the first inequality follows from Jensen's inequality. This proves that S is $(M + \frac{\eta}{2R})$-Lipschitz.

Fourth item: We need to show that for all t, $\hat{f}_t = S(x_t)$ and $\hat{g}_t = \nabla S(x_t)$. By definition, $\hat{f}_t = (F_t)_r(x_t)$ and $\hat{g}_t = \nabla (F_t)_r(x_t)$. Moreover, by Item 3 of Lemma A.4, using F_T instead of F_t gives the same quantities, namely $\hat{f}_t = (F_T)_r(x_t)$ and $\hat{g}_t = \nabla (F_T)_r(x_t)$. We claim that for every t, F_T and $\bar{S}$ are equal inside the ball $B(x_t, \sqrt{2}\, r)$, which then implies that $\hat{f}_t = (F_T)_r(x_t) = \bar{S}_r(x_t) = S(x_t)$ and $\hat{g}_t = \nabla (F_T)_r(x_t) = \nabla \bar{S}_r(x_t) = \nabla S(x_t)$, as desired. To show the equality in the ball $B(x_t, \sqrt{2}\, r)$, it suffices that the other term in the max defining $\bar{S}$ does not "cut off" F_T, namely that $f(x) - s_{T+1} \leq F_T(x)$ for every $x \in B(x_t, \sqrt{2}\, r)$. But this follows from Item 2 of Lemma A.4.

Substituting the value $r = \sqrt{\eta/\alpha}$ and $s_t = 4\eta t + \alpha r^2 t$ in the items above concludes the proof of Theorem 4.1. □

B. Proof of Lemma 3.2

Lemma 3.2: Consider a sequence of points $x_1, \ldots, x_T \in B(R)$, and a sequence of first-order pairs $(\hat{f}_1, \hat{g}_1), \ldots, (\hat{f}_T, \hat{g}_T)$. Consider $\delta > 0$ and $M' \geq M$, and suppose that there is an M'-Lipschitz convex extension F of these first-order pairs that satisfies:

$$\underbrace{f(x) \geq F(x) - \delta}_{\text{approx. under-approximation}}, \qquad \forall x \in B(R) \tag{B.1}$$

$$\underbrace{f(x_t) \leq F(x_t) + \delta}_{\text{approx. queried values}}, \qquad \forall t = \{1, \ldots, T\}. \tag{B.2}$$

Then the first-order pairs have an M'-Lipschitz convex extension F' such that $\|F' - f\|_\infty \leq \delta$.

Proof. Define the function F' as $F'(x) := \max\{f(x) - \delta, F(x)\}$ by taking the maximum between F and a downward-shifted f. We will show that this function is the desired convex extension of the first-order pairs $(\hat{f}_1, \hat{g}_1), \ldots, (\hat{f}_T, \hat{g}_T)$.

First, to show $\|F' - f\|_\infty \leq \delta$, by the definition of F' one has $F'(x) \geq f(x) - \delta$ for all x. Furthermore, because of the guarantee that $F(x) \leq f(x) + \delta$, we also have $F'(x) \leq f(x) + \delta$ for all x; together these imply that $\|F' - f\|_\infty \leq \delta$. Since f and F are M'-Lipschitz convex functions, so is F'.

It remains to be shown that F' is an extension of the first-order pairs, that is, to show $\hat{f}_t = F'(x_t)$ and $\hat{g}_t \in \partial F'(x_t)$ for all $t = 1, ..., T$. Given property (B.2), we have $F(x_t) \geq f(x_t) - \delta$, and so $F'(x_t) = F(x_t) = \hat{f}_t$. The fact that $F'(x_t) = F(x_t)$ also implies that every vector in $\partial F(x_t)$ is a subgradient of F' at x_t, namely $\partial F'(x_t) \supseteq \partial F(x_t) \ni \hat{g}_t$. To see this, recall that since F is convex, for $\hat{g}_t \in \partial F(x_t)$ we have $F(x_t) + \langle \hat{g}_t, x_t - x\rangle \leq F(x)$. Using the fact that $F'(x_t) = F(x_t)$, we thus have $F'(x_t) + \langle \hat{g}_t, x_t - x\rangle \leq F(x) \leq F'(x)$ for all x, and so any $\hat{g}_t \in \partial F(x_t)$ is also a subgradient for F' at x_t, as desired to conclude the proof. □

C. Separation oracles: proofs of Theorems 2.6 and 2.7

We now consider the original constrained problem $\min\{f(x) : x \in C \cap X\}$, and show how to run any first-order algorithm $\mathcal{A}$ using only approximate first-order information about f and approximate separation information from C, proving Theorems 2.6 and 2.7. The main additional element is to convert the approximate separation information for C into an exact information for a related set $K \approx C$ so it can be used in a black-box fashion by $\mathcal{A}$, as the previous section did for the first-order information of f. For simplicity, we assume throughout that the algorithm $\mathcal{A}$ only queries points in $B(R)$ (the ball containing the feasible set C), since points outside it can be separated exactly.

Given a set of points $x_1, ..., x_T \in B(R)$, we say a sequence of separation responses $r_1, ..., r_T \in \{\textsc{Feasible}, \textsc{Infeasible}\} \times \mathbb{R}^d$ has a *convex extension* if there is a convex set $K \neq \emptyset$ such that there exists an exact (i.e., 0-approximate) separation oracle for K giving responses $r_1, ..., r_T$ for the query points $x_1, ..., x_T$. We will also refer to such responses as *consistent with* K. As in the previous section, responses from an η-approximate separation oracle may not by themselves admit a convex extension, and need to be modified in order to allow a consistent, convex extension; for example, approximate separating hyperplanes may not be consistent with a convex set, or may "cut off" points previously reported as feasible. When we say a point y is cut off by a separating hyperplane through x with normal vector g, we mean that $\langle g, y\rangle > \langle g, x\rangle$, i.e. that y is not in the induced halfspace. Note that when given an exact separating hyperplane for some $x \notin C$, no point in C is cut off by it, whereas approximate separating hyperplanes have no such guarantee. With this in mind, we now give a theorem serving as a feasibility analogue to Theorem 3.1.

Definition C.1. For any convex set $C \subseteq \mathbb{R}^d$ and any $\delta > 0$, we define $C_{-\delta} := \{x \in C : B(x, \delta) \subseteq C\}$, which will be called *$\delta$-deep points of* C.

Theorem C.2 (Online Convex Extensibility)**.** Consider a convex set $C \subseteq B(R)$ and a sequence of points $x_1, \ldots, x_T \in B(R)$. There is an online procedure that, given access to an η-approximate separation oracle for C, produces separation responses $\hat{r}_1, ..., \hat{r}_T$ that have a convex extension K satisfying $C_{-\eta} \subseteq K \subseteq C$. Moreover, the procedure only probes the approximate oracle at the points $x_1, \ldots, x_T$.

Note that the guarantee $C_{-\eta} \subseteq K \subseteq C$ means that for any point x_t that is η_C-deep in C, i.e., in $C_{-\eta}$, the response $\hat{r}_t$ produced says $\textsc{Feasible}$, whereas for any $x_t \notin C$ it says $\textsc{Infeasible}$ and gives a hyperplane separating x_t from K (which cannot cut too deep into C, i.e., it contains $C_{-\eta}$). At a high-level, such responses allow one to cut off infeasible solutions, but guarantee that there are still (η_C-deep) solutions with small f-value available.

With this additional procedure at hand, we extend Procedure 1 from the main text in the following way to solve constrained optimization: in each step of the procedure, we also send the point x_t queried by the algorithm $\mathcal{A}$ to the η_C-approximate separation oracle for C, receive the response $\tilde{r}_t$, pass it through Theorem C.2 to obtain the new response $\hat{r}_t$, and send the latter back to $\mathcal{A}$. We call this procedure **Approximate-to-Exact-Constr**, and formally state it as follows:

Procedure 4. Approximate-to-Exact-Constr$(\mathcal{A}, T)$

For each timestep $t = 1, \ldots, T$:

1. Receive query point $x_t \in B(R)$ from $\mathcal{A}$
2. Send point x_t to the η_f-approximate first-order oracle and to the η_C-approximate separation oracle, and receive the approximate first-order information $(\tilde{f}_t, \tilde{g}_t) \in \mathbb{R} \times \mathbb{R}^d$, and separation response $\tilde{r}_t \in \{\textsc{Feasible}\} \cup \{\textsc{Infeasible}\} \times \mathbb{R}^d$.
3. Use the online procedures from Theorems 3.1 and C.2 to construct the first-order pair $(\hat{f}_t, \hat{g}_t)$ and separation response $\hat{r}_t$.
4. Send $(\hat{f}_t, \hat{g}_t), \hat{r}_t$ to the algorithm $\mathcal{A}$.

Return the point returned by $\mathcal{A}$.

The proof that this procedure yields Theorem 2.6 is analogous to the one for the unconstrained case of Theorem 2.2, so we only sketch it to avoid repetition.

Proof sketch of Theorem 2.6. Let F and K be the Lipschitz and convex extensions to the answers sent to $\mathcal{A}$ that are guaranteed by Theorems 3.1 and C.2, respectively. **Approximate-to-Exact-Constr** has the same effect as $\mathcal{A}$ running on the instance (F, K). One can show that this instance belongs to $\mathcal{I}(M + \frac{\eta_f}{2R}, R, \rho - \eta_C)$. Then if $err(\cdot)$ is the error guarantee of $\mathcal{A}$ as in the statement of the theorem, this ensures that we return a solution $\bar{x} \in K \cap X$ satisfying

$$F(\bar{x}) \leq \text{OPT}(F, K) + err(T, M + \frac{\eta_f}{2R}, R, \rho - \eta_C),$$

where $\text{OPT}(F, K) := \text{argmin}\{F(x) : x \in K \cap X\}$. Since $C_{-\eta_C}$ contains a solution x with value $f(x) \leq \text{OPT}(f, C) + \frac{2MR\eta_C}{\rho}$ (e.g., Lemma 4.7 of (Basu, 2023)), we have $\text{OPT}(f, K) \leq \text{OPT}(f, C) + \frac{2MR\eta_C}{\rho}$. Finally, using the guarantee $\|F - f\|_\infty \leq 2\eta_f T$, we obtain that $f(\bar{x}) \leq \text{OPT}(f, C) + err(T, M + \frac{\eta_f}{2R}, R, \rho - \eta_C) + 4\eta_f T + \frac{2MR\eta_C}{\rho}$, concluding the proof of the theorem. □

The proof of Theorem 2.7 follows effectively the same reasoning as for Theorem 2.6; we also sketch it here. The main difference is that one needs to ensure the optimal solution x^* of 2.1 is contained in contained in the auxiliary feasible region K the algorithm $\mathcal{A}$ uses; otherwise the additional restrictions imposed by Z may lead to arbitrarily bad solutions, or even $K \cap Z$ being empty (consider for example the case of Z being a singleton on the boundary of C that is then cut off by an approximate separation response). However, since K is guaranteed to contain $C_{-\eta_C}$ and we assume that $\eta_C \leq \rho$, the fact that x^* is ρ-deep in C implies that it is also in K.

Proof sketch of Theorem 2.7. Again, let F and K be the Lipschitz and convex extensions as in the previous proof, so that the instance (F, K) belongs to $\mathcal{I}(M + \frac{\eta_f}{2R}, R, \rho - \eta_C)$ and $\mathcal{A}$ returns a solution $\bar{x} \in K \cap Z$ satisfying $F(\bar{x}) \leq \text{OPT}(F, K) + err(T, M + \frac{\eta_f}{2R}, R, \rho - \eta_C)$, where $\text{OPT}(F, K) := \text{argmin}\{F(x) : x \in K \cap Z\}$. Recall that Z is assumed to be given and known by the algorithm. Since the optimal solution for (f, C) is assumed to be in $C_{-\rho}$, and K contains $C_{-\eta_C} \supseteq C_{-\rho}$, K contains the optimal solution to to the true instance, $x^* : f(x*) = \text{OPT}(f, C)$. Finally, using the guarantee $\|F - f\|_\infty \leq 2\eta_f T$, we obtain that $f(\bar{x^*}) \leq \text{OPT}(f, C) + err(T, M + \frac{\eta_f}{2R}, R, \rho - \eta_C) + 4\eta_f T$, concluding the proof of the theorem. □

C.1. Computing convex-extensible separation responses

We now prove Theorem C.2. The result requires the existence of a convex extension K for the responses $\hat{r}_t$ that we construct, and we need $C_{-\eta} \subseteq K \subseteq C$. We provide a procedure that produces the responses $\hat{r}_1, \ldots, \hat{r}_T$ together with sets $K_1, \ldots, K_T$ so that $K_t \cap C$ is consistent with the responses up to this round, i.e., $\hat{r}_1, \ldots, \hat{r}_t$, and is sandwiched between $C_{-\eta}$ and C. The set K_t will consist of all the points that were not excluded by the separating hyperplanes of the responses up to this round. Thus, our main task is to ensure that as K_t evolves, it does not exclude the points x_τ that the responses up to now have reported as Feasible (ensuring consistency with previous responses). We also want to ensure that none of the deep points $C_{-\eta}$ is excluded.

Before stating the formal procedure, we give some intuition on how this is accomplished. Suppose one has K_{t-1} satisfying the desired properties. One receives a new point x_t and separation response $\tilde{r}_t$ from the approximate oracle, and we need to construct a response $\hat{r}_t$ and an updated set K_t to maintain the desired properties.

Suppose $\tilde{r}_t$ reports that x_t is Feasible. Our procedure ignores this information, keeps $K_t = K_{t-1}$ and creates a response $\hat{r}_t$ that is Feasible if and only if $x_t \in K_t = K_{t-1}$ (also sending a hyperplane separating x_t from K_t if $x_t \notin K_t$; notice that since this hyperplane does not cut into K_{t-1}, we do not need to update this set). Notice that $K_t \cap C$ is indeed consistent with the response $\hat{r}_t$.

The interesting case is when $\tilde{r}_t$ reports that x_t is Infeasible (so $x_t \notin C$) but $x_t \in K_{t-1}$. Thus, x_t cannot belong to $K_t \cap C$ (recall we will construct $K_t \subseteq K_{t-1}$), and so to ensure consistency our response $\hat{r}_t$ needs to report Infeasible and a separating hyperplane that excludes x_t. The first idea is to simply use separating hyperplane reported by the approximate oracle. But this can exclude points that were deemed Feasible by our previous responses $\hat{r}_\tau$ (we call these points $\overline{\text{Feas}}_{t-1}$), which would violate consistency. Thus, we first rotate this hyperplane as little as possible such that it contains all points in $\overline{\text{Feas}}_{t-1}$, and report this rotated hyperplane H (adding it to K_{t-1} to obtain K_t). While this rotation protects the points $\overline{\text{Feas}}_{t-1}$, we also need to argue that it does not stray too much away from the original approximate separating hyperplane so as to not cut into $C_{-\eta}$.

We now describe the formal procedure in detail. We use $H(g, \bar{x}) := \{x : \langle g, x \rangle \leq \langle g, \bar{x} \rangle\}$ to denote the halfspace with normal g passing through the point $\bar{x}$.

Procedure 5.

Initialize $K_0 = \mathbb{R}^d$ and $\overline{\text{Feas}}_0 = \emptyset$. For each $t = 1, \ldots, T$:

1. Query the η-approximate feasibility oracle for C at x_t, receiving the response $(flag_t, \tilde{g}_t)$
2. If $flag_t$ = Feasible and $x_t \in K_{t-1}$. Define the response $\hat{r}_t = (\textsc{Feasible}, \star)$, and set $K_t = K_{t-1}$ and $\overline{\text{Feas}}_t = \overline{\text{Feas}}_{t-1} \cup x_t$.
3. ElseIf $flag_t$ = Feasible but $x_t \notin K_{t-1}$. Set $\hat{g}_t$ be any unit vector such that the halfspace $H(\hat{g}_t, x_t)$ contains K_{t-1}. Define the response $\hat{r}_t = (\textsc{Infeasible}, \hat{g}_t)$. Set $K_t = K_{t-1}$ and $\overline{\text{Feas}}_t = \overline{\text{Feas}}_{t-1}$.

4. Else (so $flag_t = \textsc{Infeasible}$). Let $\hat{g}_t$ be a unit vector such that the induced halfspace $H(\hat{g}_t, x_t)$ contains $\overline{\text{FEAS}}_{t-1}$ that is the closest to $\tilde{g}_t$ with this property, i.e.

$$\|\hat{g}_t - \tilde{g}_t\|_2 = \min_{g \in B(1)} \left\{ \|g - \tilde{g}_t\|_2 : H(g, x_t) \supseteq \overline{\text{FEAS}}_{t-1} \right\}.$$

Define the response $\hat{r}_t = (\textsc{Infeasible}, \hat{g}_t)$. Set $K_t = K_{t-1} \cap H(\hat{g}_t, x_t)$ and $\overline{\text{FEAS}}_t = \overline{\text{FEAS}}_{t-1}$.

We remark that in Line 4, there indeed exists a halfspace supported at x_t that contains all points in $\overline{\text{FEAS}}_{t-1}$: in this case $x_t \notin C$ (since $flag_t = \textsc{Infeasible}$) and by definition $\overline{\text{FEAS}}_{t-1} \subseteq C$, so any halfspace separating x_t from C will do.

Lemma C.3. For every $t = 1, \ldots, T$, the set $K_t \cap C$, with K_t computed by Procedure 5, is a convex extension to the responses $\hat{r}_1, \ldots, \hat{r}_t$. Moreover, all these sets satisfy $K_t \supseteq C_{-\eta}$.

Proof. It suffices to prove this for each iteration, so suppose K_{t-1} with responses $\hat{r}_1, ..., \hat{r}_{t-1}$ satisfy the lemma. If Lines 2 or 3 of the procedure were executed, it is straightforward to see K_t satisfies the lemma. If Line 4 executes, $x_t \notin C$, and so the response $\hat{r}_t$ is consistent with $K_t \cap C$ since $K_t = K_{t-1} \cap H(\hat{g}_t, x_t)$. As $\overline{\text{FEAS}}_{t-1} \subseteq K_t$ by construction and $K_t \subseteq K_{t-1}$, $K_t \cap C$ is consistent with all responses $\hat{r}_1, ..., \hat{r}_t$ made. It remains to show that K_t contains $C_{-\eta}$, for which showing $H(\hat{g}_t, x_t)$ contains it suffices. Notice that since there exists an exact halfspace $H(g, x_t)$ separating $x_t \notin C$ that is $\frac{\eta}{4R}$-close to $\tilde{g}_t$ (due to the η-approximate oracle), and $H(g, x_t)$ contains C and thus $\overline{\text{FEAS}}_{t-1}$, we have $\|\hat{g}_t - \tilde{g}_t\|_2 \leq \frac{\eta}{4R}$. The triangle inequality then reveals $\|\hat{g}_t - g\|_2 \leq \|\hat{g}_t - \tilde{g}_t\|_2 + \|\tilde{g}_t - g\|_2 \leq \frac{\eta}{2R}$, and then it is easy to see that $H(\hat{g}_t, x_t)$ contains $C_{-\eta}$, concluding the proof. □