SIAM J. OPTIM.
Vol. 23, No. 1, pp. 213–240

STOCHASTIC CONVEX OPTIMIZATION WITH BANDIT FEEDBACK*

ALEKH AGARWAL†, DEAN P. FOSTER‡, DANIEL HSU§, SHAM M. KAKADE§, AND ALEXANDER RAKHLIN‡

Abstract. This paper addresses the problem of minimizing a convex, Lipschitz function f over a convex, compact set $\mathcal{X}$ under a stochastic bandit (i.e., noisy zeroth-order) feedback model. In this model, the algorithm is allowed to observe noisy realizations of the function value $f(x)$ at any query point $x \in \mathcal{X}$. The quantity of interest is the regret of the algorithm, which is the sum of the function values at algorithm's query points minus the optimal function value. We demonstrate a generalization of the ellipsoid algorithm that incurs $\widetilde{\mathcal{O}}(\text{poly}(d)\sqrt{T})$ regret. Since any algorithm has regret at least $\Omega(\sqrt{T})$ on this problem, our algorithm is optimal in terms of the scaling with T.

Key words. derivative-free optimization, bandit optimization, ellipsoid method

AMS subject classifications. 90C56, 90C25, 68T05

DOI. 10.1137/110850827

1. Introduction. Zeroth-order or derivative-free optimization concerns the optimization of an objective, given access only to function evaluations at desired query points. Since these problems arise across many disciplines, there is a rich history of literature in this area. We point the interested reader to Chapter 7 of the book [16] or the more recent text [7] and the references therein for the relevant background. Recently and somewhat independently, these problems have received increased attention from the statistics and theoretical computer science communities, due to natural applications in decision making under limited feedback; some canonical examples are network routing and Internet ad display from a pool of choices in order to maximize revenue. In this literature, the zeroth-order feedback model has been termed "bandit feedback," with emphasis on somewhat different performance measures owing to the sequential nature of the problems. We start by describing this bandit feedback model before formally stating the problem we study in this paper.

The classical multiarmed bandit problem, formulated by Robbins in 1952, is arguably the most basic setting of sequential decision making under uncertainty. Upon choosing one of k available actions ("arms"), the decision maker observes an independent realization of the arm's cost drawn according to a distribution associated with the arm. The performance of an allocation rule (algorithm) in sequentially choosing the arms is measured by *regret*, that is, the difference between the expected costs of the chosen actions and the expected cost of the best action. Various extensions of

*Received by the editors October 10, 2011; accepted for publication (in revised form) October 23, 2012; published electronically February 14, 2013. Part of this work was done while AA and DH were at the University of Pennsylvania. An extended abstract containing the main result appeared in the proceedings of the NIPS 2011 conference.

http://www.siam.org/journals/siopt/23-1/85082.html

†Microsoft Research, New York, NY 10016 (alekha@microsoft.com). The research of this author was partially supported by MSR and Google Ph.D. fellowships.

‡Department of Statistics, University of Pennsylvania, Philadelphia, PA 19104 (dean.foster@gmail.com, rakhlin@wharton.upenn.edu). The research of the fifth author was supported by NSF grant CAREER DMS-0954737.

§Microsoft Research, New England, Cambridge, MA 02142 (dahsu@microsoft.com, skakade@microsoft.com). The research of the third author was partially supported by grants AFOSR FA9550-09-1-0425, NSF IIS-1016061, and NSF IIS-713540.

the classical formulation have received much attention in recent years. In particular, research has focused on the development of optimal and efficient algorithms for multi-armed bandits with large or even infinite action spaces, relying on various assumptions on the *structure* of costs (rewards) over the action space. When such a structure is present, the information about the cost of one arm propagates to other arms as well, making the problem tractable. For instance, the mean cost function is assumed to be linear in the paper [9], facilitating global "sharing of information" over a compact convex set of actions in a d-dimensional space. A Lipschitz condition on the mean cost function allows a local propagation of information about the arms, as costs cannot change rapidly in a neighborhood of an action. This has been exploited in a number of works, notably [2, 13, 14]. Instead of the Lipschitz condition, Srinivas et al. [19] exploit the structure of Gaussian processes, focusing on the notion of the effective dimension. These various "nonparametric" bandit problems typically suffer from the curse of dimensionality, that is, the best possible convergence rates (after T queries) are typically of the form T^α, with the exponent α approaching 1 for large dimension d.

The question addressed in the present paper is, How can we leverage *convexity* of the mean cost function as a structural assumption? The main contribution of the paper is an algorithm which achieves, with high probability, an $\tilde{O}(\text{poly}(d)\sqrt{T})$ regret after T requests. This result holds for all convex Lipschitz mean cost functions. We remark that the rate does not deteriorate with d (except in the multiplicative term), implying that convexity is a strong structural assumption which turns "nonparametric" Lipschitz problems into "parametric" ones. Nevertheless, convexity is a very natural and basic assumption, and applications of our method are therefore abundant. Let us also remark that $\Omega(\sqrt{dT})$ lower bounds have been shown for linear mean cost functions [9], making our algorithm optimal up to factors polynomial in the dimension d and logarithmic in the number of iterations T.

We note that our work focuses on the so-called *stochastic* bandits setting, where the observed costs of an action are independent draws from a fixed distribution. A parallel line of literature focuses on the more difficult adversarial setting where the costs of actions change arbitrarily from round to round. Leveraging structure in nonstochastic bandit settings is more complex and is not a goal of this paper.

We start by defining some notation and the problem setup below. The next section will survey prior works and describe their connections with our work. Section 4 gives the algorithm and analysis for the special case of univariate optimization. The algorithm and analysis for higher dimensions are given in section 5.

Notation and setup. Let $\mathcal{X}$ be a compact and convex subset of $\mathbb{R}^d$, and let $f\colon \mathcal{X} \to \mathbb{R}$ be a 1-Lipschitz convex function on $\mathcal{X}$, so $|f(x) - f(x')| \leq \|x - x'\|$ for all $x, x' \in \mathcal{X}$. We assume that $\mathcal{X}$ is specified in a way so that an algorithm can efficiently construct an approximation to smallest Euclidean ball containing the set (for instance, a separation oracle suffices). Furthermore, we assume the algorithm has noisy black-box access to f. Specifically, the algorithm is allowed to query the value of f at any $x \in \mathcal{X}$, and the response to the query x is

$$y = f(x) + \varepsilon$$

where ε is an independent σ-subgaussian random variable with mean zero: $\mathbb{E}[\exp(\lambda\varepsilon)] \leq \exp(\lambda^2\sigma^2/2)$ for all $\lambda \in \mathbb{R}$. The algorithm incurs a cost $f(x)$ for each query x. The goal of the algorithm is to minimize its *regret*: after making T queries $x_1, \ldots, x_T \in \mathcal{X}$, the regret of the algorithm is

$$R_T = \sum_{t=1}^{T} \big(f(x_t) - f(x^*)\big),$$

where x^* is a minimizer of f over $\mathcal{X}$. (We do not require uniqueness of x^*.)

Since we observe noisy function values, our algorithms will make multiple queries of f at the same point. We will construct an average and confidence interval (CI) around the average for the function values at points queried by the algorithm. We will use the notation $\mathrm{LB}_{\gamma_i}(x)$ and $\mathrm{UB}_{\gamma_i}(x)$ to denote the lower and upper bounds of a CI of width γ_i for the function estimate of a point x. We will say that CIs at two points are γ-separated if $\mathrm{LB}_{\gamma_i}(x) \geq \mathrm{UB}_{\gamma_i}(y) + \gamma$ or $\mathrm{LB}_{\gamma_i}(y) \geq \mathrm{UB}_{\gamma_i}(x) + \gamma$.

2. Related work. *Asymptotic* rates of $\mathcal{O}(\sqrt{T})$ have been previously achieved by Cope [8] for unimodal functions under stringent conditions (smoothness and strong convexity of the mean cost function, in addition to the unconstrained optimum being achieved inside the constraint set). The method employed by the author is a variant of the classical Kiefer–Wolfowitz procedure [12] for estimation of an optimum point. Further, the rate $\tilde{O}(\sqrt{T})$ has been achieved in Auer, Ortner, and Szepesvári [3] for a one-dimensional nonconvex problem with finite number of optima. The result assumes continuous second derivatives of the mean function, not vanishing at the optimum, while the first derivative is assumed to be zero at the optima. The method is based on discretizing the interval and does not exploit convexity. Yu and Mannor [20] recently studied unimodal bandits, but they only consider one-dimensional and graph-structured settings. Bubeck et al. [5] consider the general setup of $\mathcal{X}$-armed bandits with Lipschitz mean cost functions and their algorithm does give $\mathcal{O}(c(d)\sqrt{T})$ regret for a dimension dependent constant $c(d)$ in some cases when the problem has a near-optimality dimension of 0. However, not all convex, Lipschitz functions satisfy this condition, and $c(d)$ can grow exponentially in d even in these special cases.

The case of convex, Lipschitz cost functions has also been looked at in the harder adversarial model [10, 13] by constructing one-point gradient estimators. However, the best-known regret bounds for these algorithms are $\mathcal{O}(T^{3/4})$. Agarwal, Dekel, and Xiao [1] show a regret bound of $\mathcal{O}(\sqrt{T})$ in the adversarial setup, when two evaluations of the same function are allowed, instead of just one. However, this does not include the stochastic bandit optimization setting since each function evaluation in the stochastic case is corrupted with independent noise, violating the critical requirement of a bounded gradient estimator that their algorithm exploits. Indeed, applying their result in our setup yields a regret bound of $\mathcal{O}(T^{3/4})$.

A related line of work attempts to solve convex optimization problems by instead posing the problem of finding a feasible point from a convex set. Different oracle models of specifying the convex set correspond to different optimization settings. The bandit setting is identical to finding a feasible point, given only a membership oracle for the convex set. Since we get only noisy function evaluations, we in fact only have access to a noisy membership oracle. While there are elegant solutions based on random walks in the easier separation oracle model [4], the membership oracle setting has been mostly studied in the noiseless setting only and uses much more complex techniques building on the seminal work of Nemirovski and Yudin [16]. The techniques have the additional drawback that they do not guarantee a low regret since the methods often explore aggressively.

As noted in the introduction, the problem addressed in this paper is closely related to noisy zeroth-order (also called derivative-free) convex optimization, whereby the algorithm queries a point of the domain and receives a noisy value of the function. Given $\epsilon > 0$, such algorithms are guaranteed to produce an ϵ-minimizer at the end of T iterations. While the literature on stochastic optimization is vast, we emphasize that an optimization guarantee does not necessarily imply a bound on regret. We explain this point in more detail below.

Since f is convex by assumption, the average $\bar{x}_T = \frac{1}{T}\sum_{t=1}^{T} x_t$ must satisfy $f(\bar{x}_T) - f(x^*) \leq R_T/T$ (by Jensen's inequality). That is, a method guaranteeing small regret is also an optimization algorithm. The converse, however, is not necessarily true. Suppose an optimization algorithm queries T points of the domain and then outputs a candidate minimizer x_T^*. Without any assumption on the behavior of the optimization method nothing can be said about the regret it suffers over T iterations. In fact, depending on the particular setup, an optimization method might prefer to spend time querying far from the minimum of the function (that is, *explore*) and then output the solution at the last step. Guaranteeing a small regret typically involves a more careful balancing of *exploration* and *exploitation*. This distinction between arbitrary optimization schemes and *anytime* methods is discussed further in the paper [18].

We note that most of the existing approaches to derivative-free optimization outlined in the recent book [7] typically search for a descent or sufficient descent direction and then take a step in this direction. However, most convergence results are asymptotic and do not provide concrete rates even in an optimization error setting. The main emphasis is often on global optimization of nonconvex functions, while we are mainly interested in convex functions in this work. Nesterov [17] analyzes schemes similar to that of Agarwal, Dekel, and Xiao [1] with access to *noiseless* function evaluations, showing $\mathcal{O}(\sqrt{dT})$ convergence for nonsmooth functions and accelerated schemes for smooth mean cost functions. However, when analyzed in a noisy evaluation setting, his rates suffer from the degradation as those of Agarwal, Dekel, and Xiao [1].

3. Outline of our approach.

The close relationship between convex optimization and the regret-minimization problem suggests a plan of attack: Check whether existing stochastic zeroth-order optimization methods (that is, methods that only query the oracle for function values) in fact minimize regret. Two types of methods for stochastic zeroth-order convex optimization are outlined in Nemirovski and Yudin [16, Chapter 9]. The first approach uses the noisy function values to estimate a gradient direction at every step and then passes this information to a stochastic first-order method. The second approach is to use the zeroth-order information to estimate function values and pass this information to a *noiseless* zeroth-order method. Nemirovski and Yudin argue that the latter approach has greater stability when compared to the former. Indeed, for a gradient estimate to be meaningful, function values should be sampled close to the point of interest, which, in turn, results in a poor quality of the estimate. This tension is also the source of difficulty in minimizing regret with a convex mean cost function.

Owing to the insights of Nemirovski and Yudin [16], we opt for the second approach, giving up the idea of estimating the first-order information. The main novel tool of the paper is a "center-point device" that allows one to quickly detect that the optimization method might be paying high regret and to act on this information. Unlike discretization-based methods, the proposed algorithm uses convexity in a crucial way. We first demonstrate the device on one-dimensional problems, where the solution is clean and intuitive. We then develop a version of the algorithm for higher dimensions, basing our construction on the beautiful zeroth-order optimization method of Nemirovski and Yudin [16]. Their method does not guarantee vanishing regret by itself, and a careful fusion of this algorithm with our center-point device is required. The overall approach would be to use the center-point device in conjunction with a modification of the classical ellipsoid algorithm.

To motivate the center-point device, consider the following situation. Suppose f is the unknown function on $\mathcal{X} = [0, 1]$, and assume for now that it is linear with a slope

$T^{-1/3}$. Let us sample function values at $x = 1/4$ and $x = 3/4$. To even distinguish the slope from a slope $-T^{-1/3}$ (which results in a minimizer on the opposite side of $\mathcal{X}$), we need $O(T^{2/3})$ points. If the function f is indeed linear, we only incur $O(T^{1/3})$ regret on these rounds. However, if instead f is a quadratic dipping between the sampled points, we incur regret of $O(T^{2/3})$. To quickly detect that the function is not flat between the two sampled points, we additionally sample at $x = 1/2$. The center point acts as a *sentinel*: if it is recognized that the function value at the center point is noticeably below the other two values, the region $[0, 1/4] \cup [3/4, 1]$ can be discarded. If it is recognized that the value of f either at $x = 1/4$ or at $x = 3/4$ is greater than others, then either $[0, 1/4]$ or $[3/4, 1]$ can be discarded. Finally, if f at all three points appears to be similar at a given scale, we have a certificate that the algorithm is not paying regret larger than this scale per query. The remaining argument proceeds similarly to the binary search or the method of centers of gravity: since a constant portion of the set is discarded every time, it only requires a logarithmic number of "cuts." We remark that the novelty is in ensuring that regret is kept small in the process; a simpler algorithm which does not query the center is sufficient to guarantee a small optimization error but incurs a large regret on the above example.

In the next section we present the algorithm that results from the above ideas for one-dimensional convex optimization. The general case in higher dimensions is presented in section 5.

4. One-dimensional case. We start with a specialization of the setting to one dimension to illustrate some of the key ideas including the center-point device. We assume without loss of generality that the domain $\mathcal{X} = [0, 1]$ and $f(x) \in [0, 1]$. (The latter can be achieved by pinning $f(x^*) = 0$ since f is 1-Lipschitz.)

4.1. Algorithm description. Algorithm 1 proceeds in a series of *epochs* demarcated by a working feasible region (the interval $[l_\tau, r_\tau]$ in epoch τ). In each epoch, the algorithm aims to discard a portion of the working feasible region determined to only contain suboptimal points. To do this, the algorithm repeatedly makes noisy queries to f at three different points in the working feasible region. Each epoch is further subdivided into *rounds*, where we query the function $(4\sigma^2 \log T)/\gamma_i^2$ times in round i at each of the points. Since the noise is σ-subgaussian by assumption, this implies that we know the function value to within γ_i with high probability (see, e.g., Lemma 4 in the paper [6]). The value γ_i is halved at every round so that the algorithm can stop the epoch with the minimal number of queries that suffice to resolve the difference between function values at any two of x_l, x_c, x_r, ensuring a low regret in each epoch. At the end of an epoch τ, the working feasible region is reduced to a subset $[l_{\tau+1}, r_{\tau+1}] \subset [l_\tau, r_\tau]$ of the current region for the next epoch $\tau + 1$, and this reduction is such that the new region is smaller in size by a constant fraction. This geometric rate of reduction guarantees that only a small number of epochs can occur before the working feasible region only contains near-optimal points.

In order for the algorithm to identify a sizable portion of the working feasible region containing only suboptimal points to discard, the queries in each epoch should be suitably chosen, and the convexity of f must be judiciously exploited. To this end, the algorithm makes its queries at three equally spaced points $x_l < x_c < x_r$ in the working feasible region.

Case 1. If the confidence intervals around $f(x_l)$ and $f(x_r)$ are sufficiently separated, then the algorithm can identify a subset of the feasible region (either to the left of x_l or to the right of x_r) that contains no near-optimal points—i.e., every point x in the subset has $f(x) \gg f(x^*)$. This subset, which is a fourth of the

Algorithm 1. One-dimensional stochastic convex bandit algorithm.

input noisy black-box access to $f\colon [0,1] \to \mathbb{R}$, total number of queries allowed T.

Let $l_1 := 0$ and $r_1 := 1$.
for epoch $\tau = 1, 2, \ldots$ **do**
  Let $w_\tau := r_\tau - l_\tau$.
  Let $x_l := l_\tau + w_\tau/4$, $x_c := l_\tau + w_\tau/2$, and $x_r := l_\tau + 3w_\tau/4$.
  for round $i = 1, 2, \ldots$ **do**
    Let $\gamma_i := 2^{-i}$.
    For each $x \in \{x_l, x_c, x_r\}$, query $f(x)$ $\frac{4\sigma^2}{\gamma_i^2}\log T$ times.
    if $\max\{\mathrm{LB}_{\gamma_i}(x_l), \mathrm{LB}_{\gamma_i}(x_r)\} \geq \min\{\mathrm{UB}_{\gamma_i}(x_l), \mathrm{UB}_{\gamma_i}(x_r)\} + \gamma_i$ **then**
      {**Case 1**: `CI's at` x_l `and` x_r `are` γ_i `separated`}
      if $\mathrm{LB}_{\gamma_i}(x_l) \geq \mathrm{LB}_{\gamma_i}(x_r)$ **then** let $l_{\tau+1} := x_l$ and $r_{\tau+1} := r_\tau$.
      if $\mathrm{LB}_{\gamma_i}(x_l) < \mathrm{LB}_{\gamma_i}(x_r)$ **then** let $l_{\tau+1} := l_\tau$ and $r_{\tau+1} := x_r$.
      Continue to epoch $\tau + 1$.
    else if $\max\{\mathrm{LB}_{\gamma_i}(x_l), \mathrm{LB}_{\gamma_i}(x_r)\} \geq \mathrm{UB}_{\gamma_i}(x_c) + \gamma_i$ **then**
      {**Case 2**: `CI's at` x_c `and` x_l `or` x_r `are` γ_i `separated`}
      if $\mathrm{LB}_{\gamma_i}(x_l) \geq \mathrm{LB}_{\gamma_i}(x_r)$ **then** let $l_{\tau+1} := x_l$ and $r_{\tau+1} := r_\tau$.
      if $\mathrm{LB}_{\gamma_i}(x_l) < \mathrm{LB}_{\gamma_i}(x_r)$ **then** let $l_{\tau+1} := l_\tau$ and $r_{\tau+1} := x_r$.
      Continue to epoch $\tau + 1$.
    end if
  end for
end for

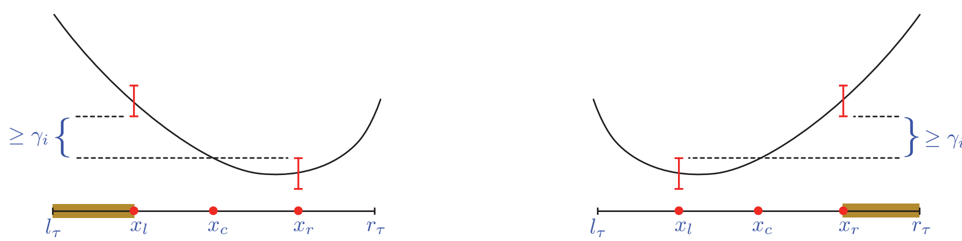

Fig. 1. *Two possible configurations when the algorithm enters Case* 1.

working feasible region by construction, is then discarded and the algorithm continues to the next epoch. This case is depicted in Figure 1.

Case 2. If the above deduction cannot be made, the algorithm looks at the confidence interval around $f(x_c)$. If this interval is sufficiently below at least one of the other intervals (for $f(x_l)$ or $f(x_r)$), then again the algorithm can identify a quartile that contains no near-optimal points, and this quartile can then be discarded before continuing to the next epoch. One possible arrangement of CIs for this case is shown in Figure 2.

Case 3. Finally, if none of the earlier cases is true, then the algorithm is assured that the function is sufficiently flat on working feasible region and hence it has not incurred much regret so far. The algorithm continues the epoch, with an increased number of queries to obtain smaller confidence intervals at each of the three points. An example arrangement of CIs for this case is shown in Figure 3.

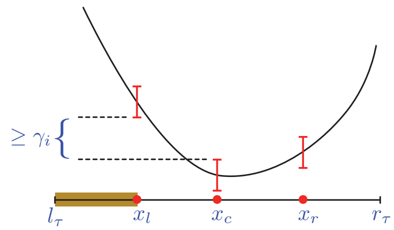

FIG. 2. *One of the possible configurations when the algorithm enters Case* 2.

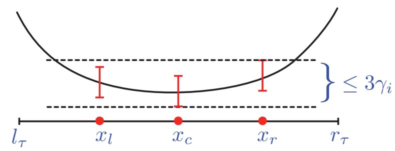

FIG. 3. *Configuration of the confidence intervals in Case* 3 *of Algorithm* 1.

4.2. Analysis. The analysis of Algorithm 1 relies on the function values being contained in the confidence intervals we construct at each round of each epoch. To avoid having probabilities throughout our analysis, we define an event $\mathcal{E}$ where at each epoch τ and each round i, $f(x) \in [\mathrm{LB}_{\gamma_i}(x), \mathrm{UB}_{\gamma_i}(x)]$ for $x \in \{x_l, x_c, x_r\}$. We will carry out the remainder of the analysis conditioned on $\mathcal{E}$ and bound the probability of $\mathcal{E}^c$ at the end.

The following theorem bounds the regret incurred by Algorithm 1. We note that the regret would be maintained in terms of the points x_t queried by the algorithm at time t.

THEOREM 1 (regret bound for Algorithm 1). *Suppose Algorithm* 1 *is run on a convex,* 1*-Lipschitz function* f *bounded in* $[0,1]$*. Suppose the noise in observations is independently and identically distributed (i.i.d.) and* σ*-subgaussian. Then with probability at least* $1 - 2/T$ *we have*

$$\sum_{t=1}^{T} f(x_t) - f(x^*) \leq 108\,\sigma\sqrt{T\log T}\log_{4/3}\left(\frac{T}{8\sigma^2\log T}\right).$$

Remarks. As stated, Algorithm 1 and Theorem 1 assume knowledge of T, but we can make the algorithm adaptive to T by a standard doubling argument. We remark that $\mathcal{O}(\sqrt{T})$ is the smallest possible regret for any algorithm even with noisy gradient information. Hence, this result shows that for purposes of regret, noisy zeroth-order information is no worse than noisy first-order information apart from logarithmic factors. We also observe that at the end of the procedure, the midpoint x_c of the working feasible region $[l_\tau, r_\tau]$, where τ was the last epoch, has an optimization error of at most $\tilde{\mathcal{O}}(1/\sqrt{T})$. This is unlike noisy first-order methods where all the iterates have to be averaged in order to get a point with low optimization error.

The theorem is proved via a series of lemmas in the next few sections. The key idea is to show that the regret on any epoch is small and the total number of epochs is bounded. To bound the per-epoch regret, we will show that the total number of queries made on any epoch depends on how close to flat the function is on the working feasible region. Thus we either take a long time but the function is very flat or we stop early when the function has sufficient slope, never accruing too much regret.

4.2.1. Bounding the regret in one epoch. We start by showing that the reduction in the working region after each epoch never discards near-optimal points.

LEMMA 1. *Suppose that the event* $\mathcal{E}$ *holds. If epoch* τ *ends in round* i*, then the interval* $[l_{\tau+1}, r_{\tau+1}]$ *contains every* $x \in [l_\tau, r_\tau]$ *such that* $f(x) \leq f(x^*) + \gamma_i$*. In particular,* $x^* \in [l_\tau, r_\tau]$ *for all epochs* τ*.*

Proof. Suppose epoch τ terminates in round i via Case 1. This means that either $\mathrm{LB}_{\gamma_i}(x_l) \geq \mathrm{UB}_{\gamma_i}(x_r) + \gamma_i$ or $\mathrm{LB}_{\gamma_i}(x_r) \geq \mathrm{UB}_{\gamma_i}(x_l) + \gamma_i$. Consider the former case. (The argument for the latter is analogous.) Since the event $\mathcal{E}$ holds, this implies that

$$f(x_l) \geq f(x_r) + \gamma_i. \tag{1}$$

Since f is convex, we can immediately conclude that every $x \in [l_\tau, l_{\tau+1}] = [l_\tau, x_l]$ has $f(x) \geq f(x^*) + \gamma_i$.

Now suppose epoch τ terminates in round i via Case 2. This means

$$\max\{\mathrm{LB}_{\gamma_i}(x_l), \mathrm{LB}_{\gamma_i}(x_r)\} \geq \mathrm{UB}_{\gamma_i}(x_c) + \gamma_i.$$

Suppose $\mathrm{LB}_{\gamma_i}(x_l) \geq \mathrm{LB}_{\gamma_i}(x_r)$. (The argument for the case $\mathrm{LB}_{\gamma_i}(x_l) < \mathrm{LB}_{\gamma_i}(x_r)$ is analogous.) The above inequality implies

$$f(x_l) \geq f(x_c) + \gamma_i.$$

We need to show that every $x \in [l_\tau, l_{\tau+1}] = [l_\tau, x_l]$ has $f(x) \geq f(x^*) + \gamma_i$. But the same argument as given in Case 1, with x_r replaced with x_c, gives the required claim.

The fact that $x^* \in [l_\tau, r_\tau]$ for all epochs τ follows by induction. ∎

The next two lemmas bound the regret incurred in any single epoch. To show this, we first establish that an algorithm incurs low regret in a round as long as it does not end an epoch. Then, as a consequence of the doubling trick, we show that the regret incurred in an epoch is on the same order as that incurred in the last round of the epoch.

LEMMA 2 (certificate of low regret). *Suppose the event $\mathcal{E}$ holds. If epoch τ continues from round i to round $i+1$, then the regret incurred in round i is at most* $\frac{144\sigma^2 \log T}{\gamma_i}$.

Proof. The regret incurred in round i of epoch τ is

$$\frac{4\sigma^2 \log T}{\gamma_i^2} \cdot \Big((f(x_l) - f(x^*)) + (f(x_c) - f(x^*)) + (f(x_r) - f(x^*)) \Big)$$

so it suffices to show that

$$f(x) \leq f(x^*) + 12\gamma_i$$

for each $x \in \{x_l, x_c, x_r\}$.

The algorithm continues from round i to round $i+1$ iff

$$\max\{\mathrm{LB}_{\gamma_i}(x_l), \mathrm{LB}_{\gamma_i}(x_r)\} < \min\{\mathrm{UB}_{\gamma_i}(x_l), \mathrm{UB}_{\gamma_i}(x_r)\} + \gamma_i$$

and

$$\max\{\mathrm{LB}_{\gamma_i}(x_l), \mathrm{LB}_{\gamma_i}(x_r)\} < \mathrm{UB}_{\gamma_i}(x_c) + \gamma_i.$$

This implies that $f(x_l)$, $f(x_c)$, and $f(x_r)$ are contained in an interval of width at most $3\gamma_i$ (recall Figure 3).

By Lemma 1, we have $x^* \in [l_\tau, r_\tau]$. Assume $x^* \leq x_c$. (The case $x^* > x_c$ is analogous.) There exists $t \geq 0$ such that $x^* = x_c + t(x_c - x_r)$, so

$$x_c = \frac{1}{1+t}x^* + \frac{t}{1+t}x_r.$$

Note that $t \le 2$ because $|x_c - l_\tau| = w_\tau/2$ and $|x_r - x_c| = w_\tau/4$, so

$$t = \frac{|x^* - x_c|}{|x_r - x_c|} \le \frac{|l_\tau - x_c|}{|x_r - x_c|} = \frac{w_\tau/2}{w_\tau/4} = 2.$$

By convexity,

$$\begin{aligned} f(x^*) &\ge (1+t)\left(f(x_c) - \frac{t}{1+t}f(x_r)\right) = f(x_r) + (1+t)\left(f(x_c) - f(x_r)\right) \\ &\ge f(x_r) - (1+t)|f(x_c) - f(x_r)| \ge f(x_r) - (1+t)\cdot 3\gamma_i \\ &\ge f(x_r) - 9\gamma_i. \end{aligned}$$

We conclude that for each $x \in \{x_l, x_c, x_r\}$,

$$f(x) \le f(x_r) + 3\gamma_i \le f(x^*) + 12\gamma_i. \qquad \square$$

LEMMA 3 (regret in an epoch). *Suppose the event $\mathcal{E}$ holds. If epoch τ ends in round i, then the regret incurred in the entire epoch is at most $\frac{432\sigma^2 \log T}{\gamma_i}$.*

Proof. If $i = 1$, then $f(x) - f(x^*) \le |x - x^*| \le 1$ for each $x \in \{x_l, x_c, x_r\}$ because f is 1-Lipschitz and $|x - x'| \le 1$ for any $x, x' \in [0, 1]$. Therefore, the regret incurred in epoch τ is

$$\frac{4\sigma^2 \log T}{\gamma_1^2} \cdot \Big((f(x_l) - f(x^*)) + (f(x_c) - f(x^*)) + (f(x_r) - f(x^*)) \Big) \le \frac{24\sigma^2 \log T}{\gamma_1}.$$

Now assume $i \ge 2$. Lemma 2 implies that the regret incurred in round j, for $1 \le j \le i-1$, is at most $\frac{144\sigma^2 \log T}{\gamma_j}$. Furthermore, for round i, we still know that the regret on each query in round i is bounded by $36\gamma_{i-1}$ ($12\gamma_{i-1}$ for each of x_l, x_c, x_r). Recalling that $\gamma_{i-1} = 2\gamma_i$ and that we make $4(\sigma^2 \log T)/\gamma_i^2$ queries at round i, the regret incurred in round i (the final round of epoch τ) is at most

$$36\gamma_{i-1}\frac{4\sigma^2 \log T}{\gamma_i^2} = \frac{288\sigma^2 \log T}{\gamma_i}.$$

Therefore, the overall regret incurred in epoch τ is

$$\begin{aligned} \sum_{j=1}^{i-1} \frac{144\sigma^2 \log T}{\gamma_j} + \frac{288\sigma^2 \log T}{\gamma_i} &= \sum_{j=1}^{i-1} 144\sigma^2 \log T \cdot 2^j + \frac{288\sigma^2 \log T}{\gamma_i} \\ &< 144\sigma^2 \log T \cdot 2^i + \frac{288\sigma^2 \log T}{\gamma_i} = \frac{432\sigma^2 \log T}{\gamma_i}. \qquad \square \end{aligned}$$

4.2.2. Bounding the number of epochs. To establish the final bound on the overall regret, we bound the number of epochs that can occur before the working feasible region only contains near-optimal points. The final regret bound is simply the product of the number of epochs and the regret incurred in any single epoch.

LEMMA 4 (bound on the number of epochs). *Suppose the event $\mathcal{E}$ holds. Then the total number of epochs τ performed by Algorithm* 1 *is bounded as*

$$\tau \le \frac{1}{2}\log_{4/3}\left(\frac{T}{8\sigma^2 \log T}\right).$$

Proof. The proof is based on observing that $\gamma_i \geq (T/4\sigma^2 \log T)^{-1/2}$ at all epochs and rounds. Indeed if $\gamma_i \leq (T/4\sigma^2 \log T)^{-1/2}$, step 7 of the algorithm would require more than T queries to get the desired confidence intervals in that round. Hence we set $\gamma_{\min} = (T/4\sigma^2 \log T)^{-1/2}$ and define the interval $I := [x^* - \gamma_{\min}, x^* + \gamma_{\min}]$ which has width $2\gamma_{\min}$. For any $x \in I$, $f(x) - f(x^*) \leq |x - x^*| \leq \gamma_{\min}$ because f is 1-Lipschitz. Moreover, for any epoch τ' which ends in round i', $\gamma_{\min} \leq \gamma_{i'}$, by definition and therefore by Lemma 1,

$$I \subseteq \{x \in [0,1] \colon f(x) \leq f(x^*) + \gamma_{i'}\} \subseteq [l_{\tau'+1}, r_{\tau'+1}].$$

This implies that $2\gamma_{\min} \leq r_{\tau'+1} - l_{\tau'+1} = w_{\tau'+1}$. Furthermore, by the definitions of $l_{\tau'+1}$, $r_{\tau'+1}$, and $w_{\tau'+1}$ in the algorithm, it follows that

$$w_{\tau'+1} \leq \frac{3}{4} \cdot w_{\tau'}$$

for any $\tau' \in \{1, \ldots, \tau\}$. Therefore, we conclude that

$$2\gamma_{\min} \leq w_{\tau+1} \leq \left(\frac{3}{4}\right)^{\tau} \cdot w_1 = \left(\frac{3}{4}\right)^{\tau},$$

which gives the claim after rearranging the inequality. ∎

4.2.3. Proof of Theorem 1. The statement of the theorem follows by combining the per-epoch regret bound of Lemma 3 with the above bound on the number of epochs and showing that all these bounds hold with sufficiently high probability.

Lemma 3 implies that the regret incurred in any epoch $\tau' \leq \tau$ that ends in round i' is at most

$$\frac{432\sigma^2 \log T}{\gamma_{i'}} \leq \frac{432\sigma^2 \log T}{\gamma_{\min}} \leq 216\sigma\sqrt{T \log T}.$$

So the overall regret incurred in all τ epochs is at most

$$216\,\sigma\sqrt{T \log T} \cdot \frac{1}{2} \log_{4/3}\left(\frac{T}{8\sigma^2 \log T}\right).$$

Finally we recall that the entire analysis thus far has been conditioned on the event $\mathcal{E}$ where all the confidence intervals we construct do contain the function values. We would now like to control the probability $\mathbb{P}(\mathcal{E}^c)$. Consider a fixed round and a fixed point x. Since the noise is σ-subgaussian, after making $4\sigma^2 \log T/\gamma_i^2$ queries we have the bound (see, e.g., Lemma 4 in [6])

$$\mathbb{P}\left(|f(x) - \hat{f}(x)| \geq \gamma_i\right) \leq \frac{2}{T^2},$$

where $\hat{f}(x)$ is the average of the observed function values. Once we have a bound for a fixed round of a fixed epoch, we would like to bound this probability uniformly over all rounds played across all epochs. We note that we make at most T queries, which is also an upper bound on the total number of rounds. Hence union bound gives

$$\mathbb{P}(\mathcal{E}^c) \leq \frac{2}{T},$$

which completes the proof of the theorem. ∎

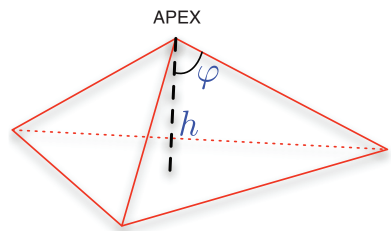

FIG. 4. *Pyramid in three dimensions.*

5. Algorithm for optimization in higher dimensions. We now move to present the general algorithm that works in d dimensions. The natural approach would be to try to generalize Algorithm 1 to work in multiple dimensions. However, the obvious extension requires constructing a covering of the unit sphere and querying the function along every direction in the covering so that we know the behavior of the function along every direction. While such an approach yields regret that scales as $\sqrt{T}$, the dependence on dimension d is exponential both in regret and the running time. The same problem was encountered in the scenario of zeroth-order optimization by Nemirovski and Yudin [16], and they use a clever construction to capture all the directions in polynomially many queries. We define a pyramid to be a d-dimensional polyhedron defined by $d+1$ points; d points form a d-dimensional regular polygon that is the base of the pyramid, and the apex lies above the hyperplane containing the base. (See Figure 4 for a graphical illustration in three dimensions.) The idea of Nemirovski and Yudin was to build a sequence of pyramids, each capturing the variation of function in certain directions, in such a way that in $\mathcal{O}(d \log d)$ pyramids we can explore all the directions. However, as mentioned earlier, their approach fails to give a low regret. We combine their geometric construction with ideas from the one-dimensional case to obtain a low-regret algorithm as described in Algorithm 2 below. Concretely, we combine the geometrical construction of Nemirovski and Yudin [16] with the center-point device to show low regret.

Just like the one-dimensional case, Algorithm 2 proceeds in *epochs*. We start with the optimization domain $\mathcal{X}$, and at the beginning we set $\mathcal{X}_0 = \mathcal{X}$. At the beginning of epoch τ, we have a current feasible set $\mathcal{X}_\tau$ which contains an approximate optimum of the convex function. The epoch ends with discarding some portion of the set $\mathcal{X}_\tau$ such that we still retain at least one approximate optimum in the remaining set $\mathcal{X}_{\tau+1}$.

At the start of the epoch τ, we start by constructing an approximation to the Löwner–John ellipsoid for the set $\mathcal{X}_\tau$, the minimum volume ellipsoid enclosing the set. While the construction of the exact Löwner–John ellipsoid is computationally intractable in general, one can use approximate construction through the ellipsoid method. (See, e.g., [16] and the discussion following Theorem 3.1 in Lovász [15].) Following the notation of Lovász [15], we call such an enclosing ellipsoid a *weak Löwner–John* ellipsoid. We next apply an affine transformation to $\mathcal{X}_\tau$ so that this ellipsoid is a Euclidean ball of radius R_τ (denoted as $\mathcal{B}(R_\tau)$). We define $r_\tau = R_\tau / c_1 d^{3/2}$ for a constant $c_1 \geq 1$, so that $\mathcal{B}(r_\tau) \subseteq \mathcal{X}_\tau$. (Such a construction is always possible; see, e.g., Theorem 3.1 in Lovász [15].) We will use the notation $\mathcal{B}_\tau$ to refer to the enclosing ball. Within each epoch, the algorithm proceeds in several rounds, each round maintaining a value γ_i which is successively halved.

Let x_0 be the center of the ball $\mathbb{B}(R_\tau)$ containing $\mathcal{X}_\tau$. At the start of a round i, we construct a regular simplex centered at x_0 and contained in $\mathbb{B}(r_\tau)$. The algorithm queries the function f at all the vertices of the simplex, denoted by $x_1 \ldots, x_{d+1}$, ALGORITHM 2. Stochastic convex bandit algorithm.

input feasible region $\mathcal{X} \subset \mathbb{R}^d$; noisy black-box access to $f\colon \mathcal{X} \to \mathbb{R}$, constants c_1 and c_2, functions $\Delta_\tau(\gamma)$, $\bar{\Delta}_\tau(\gamma)$ and number of queries T allowed.

Let $\mathcal{X}_1 := \mathcal{X}$.
for epoch $\tau = 1, 2, \ldots$ **do**
  Round $\mathcal{X}_\tau$ so $\mathbb{B}(r_\tau) \subseteq \mathcal{X}_\tau \subseteq \mathbb{B}(R_\tau)$ and $r_\tau := R_\tau/(c_1 d^{3/2})$. Let $\mathcal{B}_\tau = \mathbb{B}(R_\tau)$.
  Construct regular simplex with vertices $x_1, \ldots, x_{d+1}$ on the surface of $\mathbb{B}(r_\tau)$.
  for round $i = 1, 2, \ldots$ **do**
    Let $\gamma_i := 2^{-i}$.
    Query f at x_j for each $j = 1, \ldots, d+1$ $\frac{4\sigma^2 \log T}{\gamma_i^2}$ times.
    Let $y_1 := \arg\max_{x_j} \mathrm{LB}_{\gamma_i}(x_j)$.
    for pyramid $k = 1, 2, \ldots$ **do**
      Construct pyramid Π_k with apex y_k; let $z_1, \ldots, z_d$ be the vertices of the base of Π_k and z_0 be the center of Π_k.
      Let $\widehat{\gamma} := 2^{-1}$.
      loop
        Query f at each of $\{y_k, z_0, z_1, \ldots, z_d\}$ $\frac{4\sigma^2 \log T}{\widehat{\gamma}^2}$ times.
        Let CENTER $:= z_0$, APEX $:= y_k$, TOP be the vertex v of Π_k maximizing $\mathrm{LB}_{\widehat{\gamma}}(v)$, BOTTOM be the vertex v of Π_k minimizing $\mathrm{LB}_{\widehat{\gamma}}(v)$.
        if $\mathrm{LB}_{\widehat{\gamma}}(\text{TOP}) \geq \mathrm{UB}_{\widehat{\gamma}}(\text{BOTTOM}) + \Delta_\tau(\widehat{\gamma})$ and $\mathrm{LB}_{\widehat{\gamma}}(\text{TOP}) \geq \mathrm{UB}_{\widehat{\gamma}}(\text{APEX}) + \widehat{\gamma}$ **then**
          {Case 1(a)}
          Let $y_{k+1} :=$ TOP, and immediately continue to pyramid $k+1$.
        else if $\mathrm{LB}_{\widehat{\gamma}}(\text{TOP}) \geq \mathrm{UB}_{\widehat{\gamma}}(\text{BOTTOM}) + \Delta_\tau(\widehat{\gamma})$ and $\mathrm{LB}_{\widehat{\gamma}}(\text{TOP}) < \mathrm{UB}_{\widehat{\gamma}}(\text{APEX}) + \widehat{\gamma}$ **then**
          {Case 1(b)}
          Set $(\mathcal{X}_{\tau+1}, \mathcal{B}'_{\tau+1}) =$ CONE-CUTTING$(\Pi_k, \mathcal{X}_\tau, \mathcal{B}_\tau)$, and proceed to epoch $\tau + 1$.
        else if $\mathrm{LB}_{\widehat{\gamma}}(\text{TOP}) < \mathrm{UB}_{\widehat{\gamma}}(\text{BOTTOM}) + \Delta_\tau(\widehat{\gamma})$ and $\mathrm{UB}_{\widehat{\gamma}}(\text{CENTER}) \geq \mathrm{LB}_{\widehat{\gamma}}(\text{BOTTOM}) - \bar{\Delta}_\tau(\widehat{\gamma})$ **then**
          {Case 2(a)}
          Let $\widehat{\gamma} := \widehat{\gamma}/2$.
          if $\widehat{\gamma} < \gamma_i$ **then** start next round $i + 1$.
        else if $\mathrm{LB}_{\widehat{\gamma}}(\text{TOP}) < \mathrm{UB}_{\widehat{\gamma}}(\text{BOTTOM}) + \Delta_\tau(\widehat{\gamma})$ and $\mathrm{UB}_{\widehat{\gamma}}(\text{CENTER}) < \mathrm{LB}_{\widehat{\gamma}}(\text{BOTTOM}) - \bar{\Delta}_\tau(\widehat{\gamma})$ **then**
          {Case 2(b)}
          Set $(\mathcal{X}_{\tau+1}, \mathcal{B}'_{\tau+1})=$ HAT-RAISING$(\Pi_k, \mathcal{X}_\tau, \mathcal{B}_\tau)$, and proceed to epoch $\tau + 1$.
        end if
      end loop
    end for
  end for
end for

until the CIs at each vertex shrink to γ_i. The algorithm then picks the point y_1 for which the average of observed function values is the largest. By construction, we are guaranteed that $f(y_1) \geq f(x_j) - \gamma_i$ for all $j = 1, \ldots, d+1$. This step is depicted in Figure 5.

Algorithm 3. Cone-cutting.

input pyramid Π with apex y, (rounded) feasible region $\mathcal{X}_\tau$ for epoch τ, enclosing ball $\mathcal{B}_\tau$

1: Let $z_1, \ldots, z_d$ be the vertices of the base of Π, and $\bar{\varphi}$ the angle at its apex.

2: Define the cone

$$\mathcal{K}_\tau = \left\{ x \mid \exists \lambda > 0, \alpha_1, \ldots, \alpha_d > 0, \sum_{i=1}^{d} \alpha_i = 1 \ : \ x = y - \lambda \sum_{i=1}^{d} \alpha_i (z_i - y) \right\}.$$

3: Set $\mathcal{B}'_{\tau+1}$ to be a *weak Löwner–John* ellipsoid containing $\mathcal{B}_\tau \setminus \mathcal{K}_\tau$.

4: Set $\mathcal{X}_{\tau+1} = \mathcal{X}_\tau \cap \mathcal{B}'_{\tau+1}$.

output new feasible region $\mathcal{X}_{\tau+1}$ and enclosing ellipsoid $\mathcal{B}'_{\tau+1}$.

Algorithm 4. Hat-raising.

input pyramid Π with apex y, (rounded) feasible region $\mathcal{X}_\tau$ for epoch τ, enclosing ball $\mathcal{B}_\tau$.

1: Let CENTER be the center of Π.

2: Set $y' = y + (y - \text{CENTER})$.

3: Set Π' to be the pyramid with apex y' and same base as Π.

4: Set $(\mathcal{X}_{\tau+1}, \mathcal{B}'_{\tau+1}) = \text{Cone-cutting}(\Pi', \mathcal{X}_\tau, \mathcal{B}_\tau)$.

output new feasible region $\mathcal{X}_{\tau+1}$ and enclosing ellipsoid $\mathcal{B}'_{\tau+1}$.

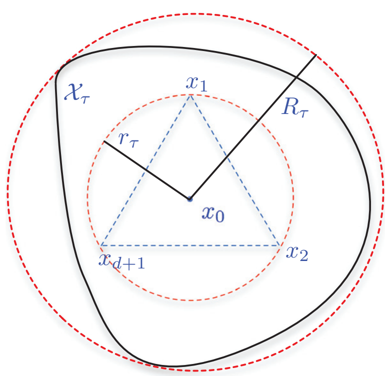

Fig. 5. *The regular simplex constructed at round i of epoch τ with radius r_τ, center x_0, and vertices $x_1, \ldots, x_{d+1}$.*

The algorithm now successively constructs a sequence of pyramids with the goal of identifying a region of the feasible set $\mathcal{X}_\tau$ such that at least one approximate optimum of f lies outside the selected region. This region will be discarded at the end of the epoch. The construction of the pyramids follows the construction from section 9.2.2 of the book [16]. The pyramids we construct will have an angle 2φ at the apex, where $\cos\varphi = c_2/d$. The base of the pyramid consists of vertices $z_1, \ldots, z_d$ such that $z_i - x_0$ and $y_1 - z_i$ are orthogonal. We note that the construction of such a pyramid is always possible—we take a sphere with $y_1 - x_0$ as the diameter, and arrange $z_1, \ldots, z_d$ on the boundary of the sphere such that the angle between $y_1 - x_0$ and $y_1 - z_i$ is φ. The construction of the pyramid is depicted in Figure 6. Given this pyramid, we set $\widehat{\gamma} = 1$ and sample the function at y_1 and $z_1, \ldots, z_d$ as well as the center of the pyramid until

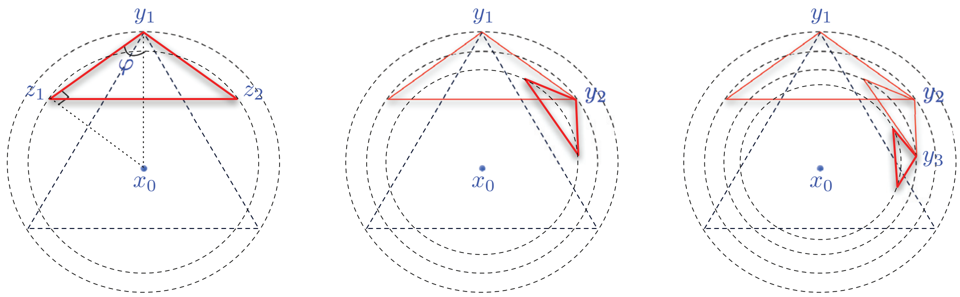

FIG. 6. *Pyramids constructed by Algorithm* 2. *First diagram is the initial pyramid constructed by the algorithm at round i of epoch τ with apex y_1, base vertices $z_1, \ldots, z_d$, and angle φ at the vertex. The other diagrams show the subsequent pyramids which successively get closer to the center.*

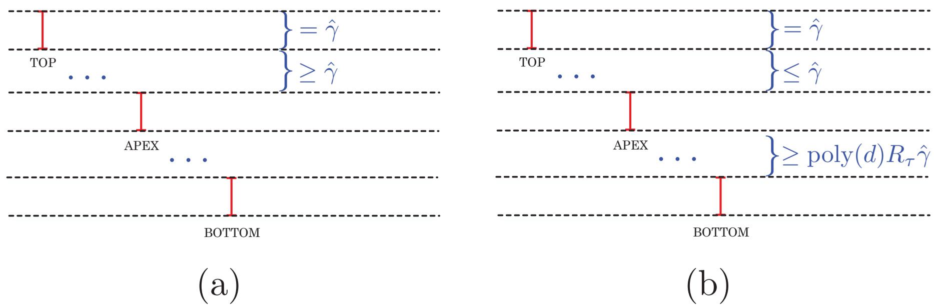

FIG. 7. *Relative ordering of confidence intervals of* TOP, BOTTOM, *and* APEX *in Cases* 1(a) *and* 1(b) *of the algorithm, respectively.*

the CIs all shrink to $\widehat{\gamma}$. Let TOP and BOTTOM denote the vertices of the pyramid (including y_1) with the largest and the smallest function value estimates, respectively. For consistency, we will also use APEX to denote the apex y_1. We then check for one of the following conditions:

1. If $\mathrm{LB}_{\widehat{\gamma}}(\text{TOP}) \geq \mathrm{UB}_{\widehat{\gamma}}(\text{BOTTOM}) + \Delta_\tau(\widehat{\gamma})$, we proceed based on the separation between TOP and apex CIs as illustrated in Figures 7(a) and 7(b).
   (a) If $\mathrm{LB}_{\widehat{\gamma}}(\text{TOP}) \geq \mathrm{UB}_{\widehat{\gamma}}(\text{APEX}) + \widehat{\gamma}$, then we know that with high probability

   $$f(\text{TOP}) \geq f(\text{APEX}) + \widehat{\gamma} \geq f(\text{APEX}) + \gamma_i. \tag{2}$$

   In this case, we set TOP to be the apex of the next pyramid, reset $\widehat{\gamma} = 1$, and continue the sampling procedure on the next pyramid.

   (b) If $\mathrm{LB}_{\widehat{\gamma}}(\text{TOP}) \leq \mathrm{UB}_{\widehat{\gamma}}(\text{APEX}) + \widehat{\gamma}$, then we know that

   $$\mathrm{LB}_{\widehat{\gamma}}(\text{APEX}) \geq \mathrm{UB}_{\widehat{\gamma}}(\text{BOTTOM}) + \Delta_\tau(\widehat{\gamma}) - 2\widehat{\gamma}.$$

   In this case, we declare the epoch over and pass the current apex to the cone-cutting step.

2. If $\mathrm{LB}_{\widehat{\gamma}}(\text{TOP}) \leq \mathrm{UB}_{\widehat{\gamma}}(\text{BOTTOM}) + \Delta_\tau(\widehat{\gamma})$, then one of the two events depicted in Figure 8(a) or 8(b) has to happen:
   (a) If $\mathrm{UB}_{\widehat{\gamma}}(\text{CENTER}) \geq \mathrm{LB}_{\widehat{\gamma}}(\text{BOTTOM}) - \bar{\Delta}_\tau(\widehat{\gamma})$, then all the vertices and the center of the pyramid have their function values within a $2\Delta_\tau(\widehat{\gamma}) + 3\widehat{\gamma}$ interval. In this case, we set $\widehat{\gamma} = \widehat{\gamma}/2$. If this sets $\widehat{\gamma} < \gamma_i$, we start the next round with $\gamma_{i+1} = \gamma_i/2$. Otherwise, we continue sampling the current pyramid with the new value of $\widehat{\gamma}$.

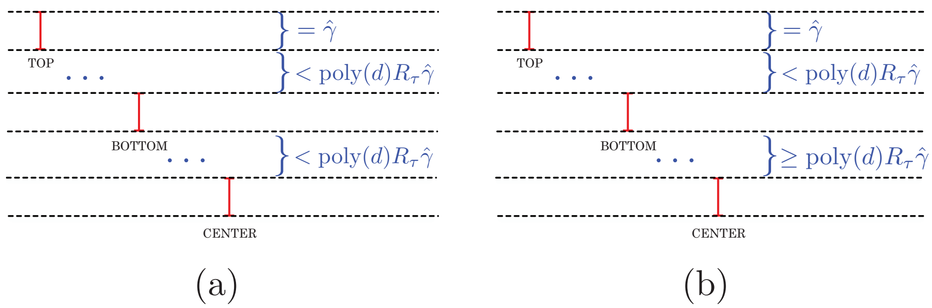

FIG. 8. *Relative ordering of confidence intervals of* TOP, BOTTOM, *and* CENTER *in Cases* 2(a) *and* 2(b) *of the algorithm, respectively.*

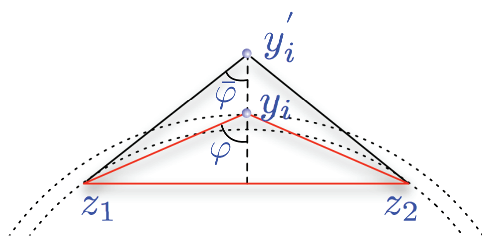

FIG. 9. *Transformation of the pyramid* Π *in the hat-raising step.*

(b) If $\text{UB}_{\widehat{\gamma}}(\text{CENTER}) \leq \text{LB}_{\widehat{\gamma}}(\text{BOTTOM}) - \bar{\Delta}_\tau(\widehat{\gamma})$, then we terminate the epoch and pass the center and the current apex to the hat-raising step.

HAT-RAISING. This step happens when we construct a pyramid where $\text{LB}_{\widehat{\gamma}}(\text{TOP}) \leq \text{UB}_{\widehat{\gamma}}(\text{BOTTOM}) + \Delta_\tau(\widehat{\gamma})$ but $\text{UB}_{\widehat{\gamma}}(\text{CENTER}) \leq \text{LB}_{\widehat{\gamma}}(\text{BOTTOM}) - \bar{\Delta}_\tau(\widehat{\gamma})$. (See Figure 8(b) for an illustration.) In this case, we will show that if we move the apex of the pyramid a little from y_i to y_i', then y_i''s CI is above the TOP CI, while the angle of the new pyramid at y_i' is not much smaller than 2φ. In particular, letting CENTER_i denote the center of the pyramid, we set $y_i' = y_i + (y_i - \text{CENTER}_i)$. Figure 9 shows transformation of the pyramid involved in this step. The correctness of this step and the sufficiency of the perturbation from y to y' will be proved in the next section.

CONE-CUTTING. This step is the concluding step for an epoch. The algorithm gets to this step either through Case 1(b) or through the hat-raising step. In either case, we have a pyramid with an apex y, a base $z_1, \dots, z_d$, and an angle $2\bar{\varphi}$ at the apex, where $\cos(\bar{\varphi}) \leq 1/2d$. We now define a cone

$$\mathcal{K}_\tau = \left\{ x \mid \exists \lambda > 0, \alpha_1, \dots, \alpha_d > 0, \sum_{i=1}^d \alpha_i = 1 \ : \ x = y - \lambda \sum_{i=1}^d \alpha_i (z_i - y) \right\} \tag{3}$$

which is centered at y and a reflection of the pyramid around the apex. By construction, the cone $\mathcal{K}_\tau$ has an angle $2\bar{\varphi}$ at its apex. We set $\mathcal{B}_{\tau+1}'$ to be a *weak Löwner–John* ellipsoid containing $\mathcal{B}_\tau \setminus \mathcal{K}_\tau$ and define $\mathcal{X}_{\tau+1} = \mathcal{X}_\tau \cap \mathcal{B}_{\tau+1}'$. This is illustrated in Figure 10. Finally, we put things back into an isotropic position and $\mathcal{B}_{\tau+1}$ is the ball containing $\mathcal{X}_{\tau+1}$ is in the isotropic coordinates, which is just obtained by applying an affine transformation to $\mathcal{B}_{\tau+1}'$.

Let us end the description with a brief discussion regarding the computational aspects of this algorithm. It is clear that the most computationally intensive steps

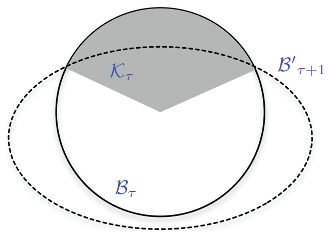

FIG. 10. *Illustration of the cone-cutting step at epoch τ. Solid circle is the enclosing ball $\mathcal{B}_\tau$. Shaded region is the intersection of $\mathcal{K}_\tau$ with $\mathcal{B}_\tau$. The dotted ellipsoid is the new enclosing ellipsoid $\mathcal{B}'_{\tau+1}$ for the residual domain.*

of this algorithm are the cone-cutting and isotropic transformation at the end. In particular, the cone-cutting step requires the construction of a weak Löwner–John ellipsoid, which can be done in polynomial time using the ellipsoid algorithm, as remarked earlier. Scaling this outer ellipsoid down by a factor of $2d^{3/2}$ yields a concentric ellipsoid fully contained in $\mathcal{X}_{\tau+1}$, allowing us to proceed to the next epoch.

6. Analysis. We start by showing the correctness of the algorithm and then proceed to regret analysis. To avoid having probabilities throughout our analysis, we define an event $\mathcal{E}$ where at each epoch τ and each round i, $f(x) \in [\mathrm{LB}_{\gamma_i}(x), \mathrm{UB}_{\gamma_i}(x)]$ for any point x sampled in the round. We will carry out the remainder of the analysis conditioned on $\mathcal{E}$ and bound the probability of $\mathcal{E}^c$ at the end. We also assume that the algorithm is run with the settings

$$\Delta_\tau(\gamma) = \left(\frac{6c_1 d^4}{c_2^2} + 3\right)\gamma \quad \text{and} \quad \bar{\Delta}_\tau(\gamma) = \left(\frac{6c_1 d^4}{c_2^2} + 5\right)\gamma \tag{4}$$

and constants $c_1 \geq 64$, $c_2 \leq 1/32$.

6.1. Correctness of the algorithm. In order to complete the proof of our algorithm's correctness, we only need to further show that when the algorithm proceeds to cone-cutting via Case 1(b), then it does not discard all the approximate optima of f by mistake, and we show that the hat-raising step is indeed correct as claimed. These two claims are established in the next couple of lemmas.

For these two lemmas, we assume that the distance of the apex of any Π constructed in epoch τ from the center of $\mathbb{B}(r_\tau)$ is at least r_τ/d. This assumption will be established later.

LEMMA 5. *Assume the event $\mathcal{E}$ holds. Let $\mathcal{K}_\tau$ be the cone discarded at epoch τ which is ended through Case* (1b) *in round i. Let* BOTTOM *be the lowest CI of the last pyramid Π constructed in the epoch, and assume the distance from the apex of Π to the center of $\mathbb{B}(r_\tau)$ is at least r_τ/d. Then $f(x) \geq f(\text{BOTTOM}) + \gamma_i$ for all $x \in \mathcal{K}_\tau$.*

Proof. Consider any $x \in \mathcal{K}_\tau$. By construction, there is a point z in the base of the pyramid Π such that the apex y of Π satisfies $y = \alpha z + (1-\alpha)x$ for some $\alpha \in [0,1)$. (See Figure 11 for a graphical illustration.)

Since f is convex and z is in the base of the pyramid, we have that

$$f(z) \leq f(\text{TOP}) \leq f(y) + 3\widehat{\gamma}.$$

Also, the condition of Case 1(b) ensures

$$f(y) > f(\text{BOTTOM}) + \Delta_\tau(\widehat{\gamma}) - 2\widehat{\gamma},$$

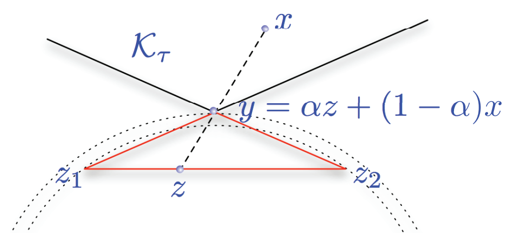

FIG. 11. *The points of interest in Lemma* 5 *(see text). Solid lines depict the pyramid* Π *and the* $\mathcal{K}_\tau$.

where $\widehat{\gamma}$ is the CI level used for the pyramid. Then by convexity of f

$$f(y) \le \alpha f(z) + (1-\alpha) f(x) \le \alpha(f(y) + 3\widehat{\gamma}) + (1-\alpha) f(x).$$

Simplifying yields

$$f(x) \ge f(y) - 3\frac{\alpha}{1-\alpha}\widehat{\gamma} > f(\text{BOTTOM}) + \Delta_\tau(\widehat{\gamma}) - 2\widehat{\gamma} - 3\frac{\alpha}{1-\alpha}\widehat{\gamma}.$$

Also, we know that $\alpha/(1-\alpha) = \|y-x\|/\|y-z\|$. Since we know that $x \in \mathbb{B}(R_\tau)$, we observe that

$$\|y-x\| \le 2R_\tau \le 2c_1\, dr_\tau.$$

Moreover, $\|y-z\|$ is at least the height of Π, which is at least $r_\tau c_2^2/d^3$ by Lemma 15. Therefore

$$\frac{\alpha}{1-\alpha} = \frac{\|y-x\|}{\|y-z\|} \le \frac{2c_1 dr_\tau}{r_\tau c_2^2/d^3} \le \frac{2c_1 d^4}{c_2^2}.$$

Thus, we have

$$f(x) > f(\text{BOTTOM}) + \Delta_\tau(\widehat{\gamma}) - 2\widehat{\gamma} - \frac{6c_1 d^4}{c_2^2}\widehat{\gamma} \ge f(\text{BOTTOM}) + \gamma_i, \tag{5}$$

where the last line uses the setting of $\Delta_\tau(\widehat{\gamma})$ (4), completing the proof of the lemma. □

This lemma guarantees that we cannot discard all the approximate minima of f by mistake in Case 1(b) and that any point discarded by the algorithm through this step in round i has regret at least γ_i. The final check that needs to be done is the correctness of the hat-raising step, which we do in the next lemma.

LEMMA 6. *Let* Π' *be the new pyramid formed in hat-raising with apex* y' *and same base as* Π *in round* i *of epoch* τ, *and let* $\mathcal{K}'_\tau$ *be the cone discarded. Assume the event* $\mathcal{E}$ *holds and that the distance from the apex of* Π *to the center of* $\mathbb{B}(r_\tau)$ *is at least* r_τ/d. *Then the* Π' *has an angle* $\bar{\varphi}$ *at the apex with* $\cos\bar{\varphi} \le 2c_2/d$, *height at most* $2r_\tau c_1^2/d^2$, *and with every point* x *in the cone* $\mathcal{K}'_\tau$ *having* $f(x) \ge f(x^*) + \gamma_i$.

Proof. Let $y' := y + (y - \text{CENTER})$ be the apex of Π'. Let h be the height of Π (the distance from y to the base), h' be the height of Π', and b be the distance from any vertex of the base to the center of the base. Then $h' < 2h \le 2r_\tau c_1^2/d^2$ by Lemma 15. Moreover, since $\cos(\varphi) = h/\sqrt{h^2+b^2} = 1/d$, we have

$$\cos(\bar{\varphi}) = h'/\sqrt{h'^2+b^2} \le 2h/\sqrt{h^2+b^2} = 2\cos(\varphi) = 2c_2/d.$$

It remains to show that every $x \in \mathcal{K}'_\tau$ has $f(x) \ge f(x^*) + \widehat{\gamma}$. By convexity of f, $f(y) \le (f(y') + f(\text{CENTER}))/2$, so $f(y') \ge 2f(y) - f(\text{CENTER})$. Since we enter hat-raising via Case 2(b) of the algorithm, we know that $f(\text{CENTER}) \le f(y) - \bar{\Delta}_\tau(\widehat{\gamma})$, so

$$f(y') \ge f(y) + \bar{\Delta}_\tau(\widehat{\gamma}).$$

The condition for entering Case 2(b) also implies that

$$f(y) > f(\text{TOP}) - \Delta_\tau(\widehat{\gamma}) - 2\widehat{\gamma} > f(x) - \Delta_\tau(\widehat{\gamma}) - 2\widehat{\gamma}$$

for all $x \in \Pi$, and therefore for any z on the base of Π,

$$f(y') > f(z) + \bar{\Delta}_\tau(\widehat{\gamma}) - \Delta_\tau(\widehat{\gamma}) - 2\widehat{\gamma} \geq f(z),$$

where the last line uses the settings of $\Delta_\tau(\widehat{\gamma})$ and $\bar{\Delta}_\tau(\widehat{\gamma})$ (4). Now take any $x \in \mathcal{K}'_\tau$. There exists $\alpha \in [0, 1)$ and z on the base of Π' such that $y' = \alpha z + (1-\alpha)x$, so by convexity of f, $f(y') \leq \alpha f(z) + (1-\alpha)f(x) \leq \alpha f(y') + (1-\alpha)f(x)$, which implies $f(x) \geq f(y') \geq f(y) + \bar{\Delta}_\tau(\widehat{\gamma}) \geq f(x^*) + \gamma_i$. ∎

6.2. Regret analysis. The following theorem states our regret guarantee on the performance of Algorithm 2.

THEOREM 2. *Assume that the convex set $\mathcal{X}$ satisfies $R_1 \leq T^{d/2}$. Suppose Algorithm 2 is run with $c_1 \geq 64$, $c_2 \leq 1/32$, and parameters*

$$\Delta_\tau(\gamma) = \left(\frac{6c_1 d^4}{c_2^2} + 3\right)\gamma \quad \textit{and} \quad \bar{\Delta}_\tau(\gamma) = \left(\frac{6c_1 d^4}{c_2^2} + 5\right)\gamma.$$

Suppose the noise in observations is i.i.d. and σ-subgaussian. Then with probability at least $1 - 2/T$, the net regret incurred by the algorithm is bounded by

$$1536\, d^{7/2}\sigma^2\sqrt{T}\log^2 T \left(\frac{2d^2\log d}{c_2^2} + 1\right)\left(\frac{4d^7 c_1}{c_2^3} + \frac{d(d+1)}{c_2}\right)\left(\frac{12c_1 d^4}{c_2^2} + 11\right).$$

Remarks. The prior knowledge of T in Algorithm 2 and Theorem 2 can again be addressed using a doubling argument. As earlier, Theorem 2 is optimal in the dependence on T. The large dependence on d is also seen in Nemirovski and Yudin [16], who obtain a d^7 scaling in the noiseless case and leave it an unspecified polynomial in the noisy case. Using random walk ideas [4] to improve the dependence on d is an interesting question for future research. We also note that the assumption $R_1 \leq T^{d/2}$ is only made for ease of presentation of the final theorem statement. A more general result in terms of R_1 easily follows from our proofs.

The analysis will start by controlling the regret incurred on different rounds, and then we will piece it together across rounds and epochs to get the net regret for the entire procedure.

6.2.1. Bounding the regret incurred in one round. We will start by a simple lemma regarding the regret incurred while playing a pyramid if condition 2(a) is encountered in the algorithm. This lemma highlights the importance of evaluating the function at the center of the pyramid, a step that was not needed in the framework of Nemirovski and Yudin [16]. We will use the symbol Π to refer to a generic pyramid constructed by the algorithm during the course of its operation, with apex y, base $z_1, \ldots, z_d$, center denoted as CENTER, and an angle φ at the apex. We also recall that the pyramids constructed by the algorithm are such that the distance from the center to the base is at least $r_\tau c_2^2/d^3$.

LEMMA 7. *Assume the event $\mathcal{E}$ holds. Suppose the algorithm reaches Case* 2(a) *in round i of epoch τ, and assume $x^* \in \mathbb{B}(R_\tau)$, where x^* is the minimizer of f. Let Π be the current pyramid and $\widehat{\gamma}$ be the current CI width. Assume the distance from the* *apex of* Π *to the center of* $\mathbb{B}(r_\tau)$ *is at least* r_τ/d. *Then the net regret incurred while evaluating the function on* Π *in round* i *is at most*

$$\frac{12d\sigma^2 \log T}{\widehat{\gamma}}\left(\frac{4d^7 c_1}{c_2^3} + \frac{d(d+2)}{c_2}\right)\left(\frac{12c_1 d^4}{c_2^2} + 11\right).$$

Proof. The proof is a consequence of convexity. We start by bounding the variation of the function inside the pyramid. Since the pyramid is a convex hull of its vertices, we know that the function value at any point in the pyramid is also upper bounded by the largest function value achieved at any vertex. Furthermore, the condition for reaching Case (2a) implies that the function value at any vertex is at most $f(\text{CENTER}) + \Delta_\tau(\widehat{\gamma}) + \bar{\Delta}_\tau(\widehat{\gamma}) + 3\widehat{\gamma}$, and therefore

$$f(x) \le f(\text{CENTER}) + \Delta_\tau(\widehat{\gamma}) + \bar{\Delta}_\tau(\widehat{\gamma}) + 3\widehat{\gamma} \quad \text{for all } x \in \Pi. \tag{6}$$

For brevity, we use the shorthand $\delta := \Delta_\tau(\widehat{\gamma}) + \bar{\Delta}_\tau(\widehat{\gamma}) + 3\widehat{\gamma}$. Consider any point $x \in \Pi$, and let b be the point where the ray $\text{CENTER} - x$ intersects a face of Π on the other side. Then we know that there is a positive constant $\alpha \in [0,1]$ such that $\text{CENTER} = \alpha x + (1-\alpha)b$; in particular, $(1-\alpha)/\alpha = \|\text{CENTER} - x\|/\|\text{CENTER} - b\|$. Note that $\|\text{CENTER} - x\|$ is at most the distance from CENTER to a vertex of Π, and $\|\text{CENTER} - b\|$ is at least the radius of the largest ball centered at CENTER inscribed in Π. Therefore by item 2 of Lemma 16,

$$\frac{1-\alpha}{\alpha} = \frac{\|\text{CENTER} - x\|}{\|\text{CENTER} - b\|} \le \frac{d(d+1)}{c_2}.$$

Then the convexity of f and the upper bound on function values over Π from (6) guarantee that

$$f(\text{CENTER}) \le \alpha f(x) + (1-\alpha) f(b) \le \alpha f(x) + (1-\alpha)(f(\text{CENTER}) + \delta).$$

Rearranging, we get

$$f(x) \ge f(\text{CENTER}) - \frac{d(d+1)\delta}{c_2}. \tag{7}$$

Combining (6) and (7) we have shown that for any $x, x' \in \Pi$

$$|f(x) - f(x')| \le \frac{d(d+2)\delta}{c_2}. \tag{8}$$

Now we will bootstrap to show that the above bound implies low regret while sampling the vertices and center of Π. We first note that if $x^* \in \Pi$, then the regret on any vertex or the center is bounded by $d(d+2)\delta/c_2$. In that case, the regret incurred by sampling the vertices and center of this pyramid (so $d+2$ points) is bounded by $(d+2) \cdot d(d+2)\delta/c_2$. Furthermore, we only need to sample each point pyramid $4\sigma^2 \log T/\widehat{\gamma}^2$ times to get the CIs of width $\widehat{\gamma}$, which completes the proof in this case, so the total regret incurred is

$$(d+2)\frac{d(d+2)\delta}{c_2} \cdot \frac{4\sigma^2 \log T}{\widehat{\gamma}^2}.$$

Now we consider the case where $x^* \notin \Pi$. Recall that Lemma 5 guarantees that $x^* \in \mathcal{B}_\tau$. There is a point b on a face of Π such that $b = \alpha x^* + (1-\alpha)\,\text{CENTER}$ for some $\alpha \in [0,1]$. Then $\alpha = \|\textsc{center} - b\| / \|\textsc{center} - x^*\|$. By the triangle inequality, $\|\textsc{center} - x^*\| \leq 2R_\tau = 2c_1 d r_\tau$. Moreover, $\|\textsc{center} - b\|$ is at least the radius of the largest ball centered at CENTER inscribed in Π, which is at least $r_\tau c_2^2/(2d^4)$ by Lemma 16. Therefore $\alpha \geq c_2^2/(4c_1 d^5)$. By convexity and (7),

$$f(\textsc{center}) - \frac{d(d+2)\delta}{c_2} \leq f(b) \leq \alpha f(x^*) + (1-\alpha) f(\textsc{center}),$$

so

$$f(x^*) \geq f(\textsc{center}) - \frac{d(d+2)\delta}{c_2 \alpha} \geq f(\textsc{center}) - \frac{4d^7 c_1 \delta}{c_2^3} \geq f(x) - \frac{4d^7 c_1 \delta}{c_2^3} - \frac{d(d+2)\delta}{c_2}$$

for any $x \in \Pi$. Therefore, using the same argument as before, the net regret incurred in the round is

$$(d+2)\left(\frac{4d^7 c_1}{c_2^3} + \frac{d(d+2)}{c_2}\right)\delta \cdot \frac{4\sigma^2 \log T}{\widehat{\gamma}^2}.$$

Substituting in the values of $\Delta_\tau(\widehat{\gamma})$ and $\bar{\Delta}_\tau(\widehat{\gamma})$ completes the proof. □

Lemma 7 is critical because it allows us to claim that at any round, when we sample the function over a pyramid with a value $\widehat{\gamma}$, the regret on that pyramid during this sampling is at most $\mathrm{poly}(d)/\widehat{\gamma}$ since we must have been in Case 2(a) with $2\widehat{\gamma}$ if we're using $\widehat{\gamma}$. The only exception is at the first round, where this statement holds trivially as the function is 1-Lipschitz by assumption.

We next show that the algorithm can visit Case 1(a) only a bounded number of times every round. The round is ended when the algorithm enters Case 1(b) or 2(b), and the regret incurred on Case 2(a) would be bounded using the above Lemma 7.

The key idea for this bound is present in section 9.2.2 of Nemirovski and Yudin [16]. We need a slight modification of their argument because the function evaluations have noise and our sampling strategy is a little different from theirs.

LEMMA 8. *Assume the event $\mathcal{E}$ holds. At any round, the number of visits to Case* 1(a) *is $2d^2 \log d / c_2^2$, and each pyramid Π constructed by the algorithm satisfies $\|y - x_0\| \geq r_\tau / d$, where y is the apex of Π.*

Proof. The proof follows by a simple geometric argument that exploits the fact that we have an angle 2φ at the apex of our pyramid which is almost equal to π and that $y - x_0$ and $z_i - x_0$ are orthogonal for any pyramid Π we construct (see Figure 6). By definition of Case 1(a), $\textsc{top} \neq y$, so we assume $\textsc{top} = z_1$ without loss of generality. By construction,

$$\|z_1 - x_0\| = \sin\varphi \|y - x_0\|. \tag{9}$$

Since this step applies every time we enter Case 1(a), the total number k of visits to Case 1(a) satisfies $\|z_1 - x_0\| = (\sin\varphi)^k r_\tau$, where we recall that r_τ is the radius of the regular simplex we construct in the first step on every round. We further note that for a regular simplex of radius r_τ, a Euclidean ball of radius r_τ/d is contained in the simplex. We also note that by construction, $\cos\varphi = c_2/d$ and hence $\sin\varphi = \sqrt{1 - c_2^2/d^2} \leq 1 - c_2^2/(2d^2)$. Hence, setting $k = 2d^2 \log d / c_2^2$ suffices to ensure that $\|z_1 - x_0\| \leq r_\tau/d$, guaranteeing that z_1 lies in the initial simplex of radius r_τ centered at x_0, as depicted in Figure 12.

Let $y_1, \ldots, y_k$ be the apexes of the pyramids we have constructed in this round. Then by construction, we have a sequence of points such that

$$f(z_1) = f(\textsc{top}) \geq f(y_k) + \gamma \geq f(y_{k-1}) + 2\gamma \cdots \geq f(y_1) + k\gamma.$$

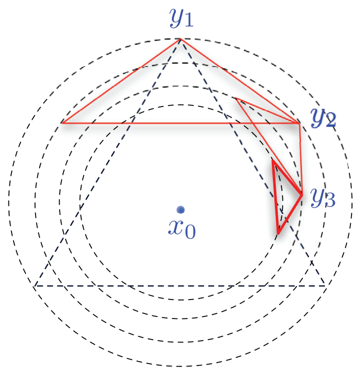

FIG. 12. *The apexes of the successive pyramids get closer to the center of the simplex* x_0 *and eventually enter the simplex after at most* $\mathcal{O}(d^2 \log d)$ *pyramids.*

On the other hand, we know that y_1 satisfies $f(y_1) \geq f(x_i) - \gamma$ for all the vertices x_i of the simplex by definition of y_1. Since z_1 lies in the simplex, convexity of f guarantees

$$f(y_1) \geq f(z_1) - \gamma \geq f(y_1) + (k-1)\gamma,$$

which is a contradiction unless $k \leq 1$. Thus it must be the case that z_1 is not in the simplex if $k > 1$, in which case k can be at most $2d^2 \log d / c_2^2$. ∎

This lemma guarantees that in at most $2d^2 \log d / c_2^2$ pyramid constructions, the algorithm will enter one of Case 1(b) or 2(b) and terminate the epoch, unless the CI level γ at this round is insufficient to resolve things and we end in Case 2(a). It also shows that all the pyramids constructed by our algorithm are sufficiently far from the center, which is assumed by Lemmas 5–7. Until now, we have focused on controlling the regret on the pyramids we construct, which is convenient since we sample the center points of the pyramids. To bound the regret incurred over one round, we also need to control the regret over the initial simplex we query at every round. We start with a lemma that shows how to control the net regret accrued over an entire round, when the round ends in Case 2(a).

LEMMA 9. *Assume the event* $\mathcal{E}$ *holds. For any round with a CI width of* γ *that terminates in Case* 2(a), *the net regret incurred on the round is at most*

$$\frac{24d\sigma^2 \log T}{\gamma} \left(\frac{2d^2 \log d}{c_2^2} + 1 \right) \left(\frac{4d^7 c_1}{c_2^3} + \frac{d(d+2)}{c_2} \right) \left(\frac{12c_1 d^4}{c_2^2} + 11 \right).$$

Proof. Suppose we constructed a total of k pyramids on the round with $k \leq 2d^2 \log d / c_2$ by Lemma 8. Then we know that the instantaneous regret on any point of the k_{th} pyramid Π_k is bounded by

$$\delta := \gamma \left(\frac{4d^7 c_1}{c_2^3} + \frac{d(d+2)}{c_2} \right) \left(\frac{12c_1 d^4}{c_2^2} + 11 \right)$$

by Lemma 7. We also note that by construction, y_k is the TOP vertex of the $(k-1)$st pyramid Π_{k-1}. Hence by definition of Case 1(a) (which caused us to go from Π_{k-1} to Π_k), we know that $f(x) \leq f(y_k) + \gamma$ for all $x \in \Pi_{k-1}$. Reasoning in the same way, we get that the function value at each vertex of the pyramid we constructed in this round is bounded by the function value at y_k. Furthermore, just like the proof of Lemma 8, the function value at any vertex of the initial simplex is also bounded by the function value at y_k. As a result, the instantaneous regret incurred at any point we sampled in this round is bounded by the net regret at y_k which is at most by δ using Lemma 7. Since every pyramid as well as the simplex samples at most $d+2$ vertices, and the total number of pyramids we construct is bounded by Lemma 8, we query at most $(d+2)(2d^2/c_2^2 \log d+1)$ points at any round. In order to bound the number of queries made at any point, we observe that for a CI level $\widehat{\gamma}$, we make $4\sigma^2 \log T/\widehat{\gamma}^2$ queries. Suppose $\gamma = 2^{-1}$. Since $\widehat{\gamma}$ is geometrically decreased to γ, the total number of queries made at any point is bounded by

$$\sum_{j=1}^{i} \frac{4\sigma^2 \log T}{2^{-2j}} \leq 16\sigma^2 \log T 2^{2i} = \frac{16\sigma^2 \log T}{\gamma^2}.$$

Putting all the pieces together, the net regret accrued over this round is at most

$$\frac{48d\sigma^2 \log T}{\gamma} \left(\frac{2d^2 \log d}{c_2^2} + 1 \right) \left(\frac{4d^7 c_1}{c_2^3} + \frac{d(d+2)}{c_2} \right) \left(\frac{12c_1 d^4}{c_2^2} + 11 \right),$$

which completes the proof. ∎

We are now in a position to state a regret bound on the net regret incurred in any round. The key idea would be to use the bound from Lemma 9 to bound the regret even when the algorithm terminates in Case 1(b) or 2(b).

LEMMA 10. *Assume the event $\mathcal{E}$ holds. For any round that terminates in a CI level γ, the net regret over the round is bounded by*

$$\frac{96d\sigma^2 \log T}{\gamma} \left(\frac{2d^2 \log d}{c_2^2} + 1 \right) \left(\frac{4d^7 c_1}{c_2^3} + \frac{d(d+2)}{c_2} \right) \left(\frac{12c_1 d^4}{c_2^2} + 11 \right).$$

Proof. We just need to control the regret incurred in rounds that end in Case 1(b) or 2(b). We recall from the description of the algorithm that a CI level of γ is used at a round only when the algorithm terminates the round with a CI level of 2γ in Case 2(a). The only exception is the first round with $\gamma = 1$, where the instantaneous regret is bounded by 1 at any point using the Lipschitz assumption. Now suppose we did end a round with CI level 2γ in Case 2(a). In particular, the proof of Lemma 9 guarantees that the instantaneous regret at any vertex of the simplex we construct is at most

$$2\gamma \left(\frac{4d^7 c_1}{c_2^3} + \frac{d(d+2)}{c_2} \right) \left(\frac{12c_1 d^4}{c_2^2} + 11 \right).$$

Now consider any pyramid constructed on this round. We know that the instantaneous regret incurred if the pyramid ends in Case 2(a) is bounded by Lemma 7. Furthermore, if the algorithm was in Case 1(a), 1(b), or 2(b) with a CI level $\widehat{\gamma}$ (which could be larger than γ in general), then it must have been in Case 2(a) with a CI level $2\widehat{\gamma}$. Hence the instantaneous regret on the vertices of the pyramid is at most

$$2\widehat{\gamma} \left(\frac{4d^7 c_1}{c_2^3} + \frac{d(d+2)}{c_2} \right) \left(\frac{12c_1 d^4}{c_2^2} + 11 \right),$$

and we make at most $\frac{16\sigma^2 \log T}{\widehat{\gamma}^2}$ queries on any point of the pyramid by a similar argument like the previous lemma. Thus the net regret incurred at any pyramid constructed by the algorithm is at most

$$\frac{96d\sigma^2 \log T}{\widehat{\gamma}} \left(\frac{4d^7 c_1}{c_2^3} + \frac{d(d+2)}{c_2}\right)\left(\frac{12c_1 d^4}{c_2^2} + 11\right).$$

Recalling our bound on the number of pyramids constructed at any round completes the proof. ∎

Putting all the pieces together, we have shown that the regret incurred on any round with a CI level γ is bounded by C/γ, where C comes from the above lemmas. We further observe that since γ is reduced geometrically, the net regret incurred on an epoch where the largest CI level we encounter is γ is at most

$$\sum_{j=1}^{i} \frac{C}{2^{-j}} \leq 2C2^i = 2C/\gamma.$$

This allows us to get a bound on the regret of one epoch stated in the next lemma.

LEMMA 11. *The regret in any epoch which ends in CI level γ is at most*

$$\frac{192d\sigma^2 \log T}{\gamma}\left(\frac{2d^2 \log d}{c_2^2} + 1\right)\left(\frac{4d^7 c_1}{c_2^3} + \frac{d(d+2)}{c_2}\right)\left(\frac{12c_1 d^4}{c_2^2} + 11\right). \tag{10}$$

6.2.2. Bound on the number of epochs. In order to bound the number of epochs, we first need to show that the cone-cutting step discards a sizeable chunk of the set $\mathcal{X}_\tau$ in epoch τ. Recall that we need to understand the ratio of the volumes of $\mathcal{B}_{\tau+1}$ to $\mathcal{B}_\tau$ in order to understand the amount of volume discarded in any epoch.

LEMMA 12. *Let $\mathcal{B}_\tau$ be the smallest ball containing $\mathcal{X}_\tau$, and let $\mathcal{B}'_{\tau+1}$ be the minimum volume ellipsoid containing $\mathcal{B}_\tau \setminus \mathcal{K}_\tau$. Then for small enough constants c_1, c_2, $vol(\mathcal{B}'_{\tau+1}) \leq \rho \cdot vol(\mathcal{B}_\tau)$ for $\rho = \exp(-\frac{1}{4(d+1)^{3/2}})$.*

Proof. This lemma is analogous to the volume reduction results proved in the analysis of the ellipsoid method for convex programming with a gradient oracle. We start by arguing that it suffices to consider the intersection of $\mathcal{B}_\tau$ with a half-space in order to understand the set $\mathcal{B}_\tau \setminus \mathcal{K}_\tau$. It is clear from the figure that we only increase the volume of the enclosing ellipsoid $\mathcal{B}'_{\tau+1}$ if we consider discarding only the spherical cap instead of discarding the entire cone. But the spherical cap is exactly obtained by taking the intersection of $\mathcal{B}_\tau$ with a half-space.

The choices of the constants c_1, c_2 earlier guarantee that the distance of the hyperplane from the origin is at most $R_\tau/(4(d+1)^{3/2})$. This is because the apex of the cone $\mathcal{K}_\tau$ is always contained in $\mathbb{B}(r_\tau)$ by construction and the height of the cone is at most $R_\tau \cos\bar{\varphi} \leq R_\tau/(8(d+1))$, where the last inequality will be ensured by construction. Ensuring $r_\tau \leq R_\tau/(32(d+1)^{3/2})$ suffices to ensure that the distance of the hyperplane to the origin is at most $R_\tau/(4(d+1)^{3/2})$.

Thus $\mathcal{B}'_{\tau+1}$ is the minimum volume ellipsoid enclosing the intersection of a sphere with a hyperplane at a distance at most $R_\tau/(4(d+1)^{3/2})$ from its center. The volume of $\mathcal{B}'_{\tau+1}$ is then bounded as stated by using Theorem 2.1 of Goldfarb and Todd [11] in their work on deep cuts for the ellipsoid algorithm. In particular, we apply their result with $\alpha = -1/(4(d+1)^{3/2})$ giving that $\mathrm{vol}(\mathcal{B}'_{\tau+1}) \leq \rho \cdot \mathrm{vol}(\mathcal{B}_\tau)$, where

$$\rho = \left(\frac{d^2}{d^2-1}\right)^{(d-1)/2} \frac{d}{d+1}(1-\alpha^2)^{(d-1)/2}(1-\alpha).$$

Noting that $1+x \leq e^x$ and $1-x \leq e^{-x}$ allows us to simplify the above expression as

$$\rho \leq \exp\left(\frac{d-1}{2}\frac{1}{d^2-1} - \frac{1}{d+1} + \frac{d-1}{2}\alpha^2 - \alpha\right).$$

Simplifying the above expression and plugging in our choice of α yields the statement of our lemma. □

We note that the connection from volume reduction to a bound on the number of epochs is somewhat delicate for our algorithm. The key idea is to show that at any epoch that ends with a CI level γ, the cone $\mathcal{K}_\tau$ contains points with regret at least γ. This will be shown in the next lemma.

LEMMA 13. *Assume that the event $\mathcal{E}$ holds. At any epoch ending with CI level γ, the instantaneous regret of any point in $\mathcal{K}_\tau$ is at least γ*

Proof. Since every epoch terminates either through Case 1(b) or through Case 2(b) followed by hat-raising, we just need to check the condition of the lemma for both cases. If the epoch proceeds to cone-cutting through Case 1(b), this is already shown in (5). Thus we only need to verify the claim when we terminate via the hat-raising step. Recall that after hat-raising, the apex y' of the final pyramid Π' constructed in the hat-raising step satisfies that $f(y') \geq f(z_i) + \gamma$ for all the vertices $z_1, \ldots, z_d$ of the pyramid. Consider any point $x \in \mathcal{K}_\tau$. This point lies on a ray from the base of Π' passing through y'. We know the function f is increasing along this ray at y' and hence continues to increase from y' to x by convexity of f, as argued in the proof of Lemma 6. Hence in this case also the instantaneous regret of any point in $\mathcal{K}_\tau$ is at least γ, completing the proof. □

The next lemma bounds the number of epochs played by the algorithm.

LEMMA 14. *Assume that $R_1 \leq T^{d/2}$ and that the event $\mathcal{E}$ holds. The total number of epochs in the algorithm is bounded by $\frac{d \log T}{\log(1/\rho)}$ with $\rho = \exp(-\frac{1}{4(d+1)^{3/2}})$.*

Proof. Let x^* be the optimum of f. Since f is 1-Lipschitz, any point in a ball of radius $1/\sqrt{T}$ centered around x^* has instantaneous regret at most $1/\sqrt{T}$. The volume of this ball is $T^{-d/2}V_d$, where V_d is the volume of a unit ball in d dimensions. Suppose the algorithm goes on for k epochs. We know that the volume of $\mathcal{X}$ after k epochs is at most $\rho^k R_1 V_d$ by Lemma 12. We also note that the instantaneous regret of any point discarded by the algorithm in any epoch is at least $1/\sqrt{T}$ using Lemma 13, since we always maintain $\gamma \geq 1/\sqrt{T}$. Thus any point in the ball of radius $1/\sqrt{T}$ around x^* is never discarded by the algorithm. As a result, the algorithm must stop once we have

$$\rho^k R_1 V_d \leq T^{-d/2} V_d,$$

which means $k \leq (d \log T/2 + \log R_1)/\log 1/\rho$. Finally, recalling that

$$\log R_1 \leq d \log T/2$$

by assumption completes the proof. □

We are now in a position to put together all the pieces.

Proof of Theorem 2. We are guaranteed that there are at most $d \log T/\log(1/\rho)$ epochs where the regret on each epoch is bounded by (10). Observing that $\gamma \geq 1/\sqrt{T}$ guarantees that every epoch has regret at most

$$192 d\sigma^2 \sqrt{T} \log T \left(\frac{2d^2 \log d}{c_2^2} + 1\right)\left(\frac{4d^7 c_1}{c_2^3} + \frac{d(d+2)}{c_2}\right)\left(\frac{12 c_1 d^4}{c_2^2} + 11\right).$$

Combining with the above bound on the number of epochs guarantees that the cumulative regret of our algorithm is bounded by

$$\frac{192d^2\sigma^2\sqrt{T}\log^2 T}{\log(1/\rho)}\left(\frac{2d^2\log d}{c_2^2}+1\right)\left(\frac{4d^7c_1}{c_2^3}+\frac{d(d+2)}{c_2}\right)\left(\frac{12c_1d^4}{c_2^2}+11\right).$$

Finally, we recall that the entire analysis this far has been conditioned on the even $\mathcal{E}$ which assumes that the function value lies in the confidence intervals we construct at every round. By design, just like the proof of Theorem 1, $\mathbb{P}(\mathcal{E}^c) \le 2/T$. Substituting the value of ρ from Lemma 14 completes the proof of the theorem. ∎

7. Discussion. This paper presents a new algorithm for convex optimization when only noisy function evaluations are possible. The algorithm builds on the techniques of Nemirovski and Yudin [16] from zeroth-order optimization. The key contribution of our work is to extend their algorithm to a noisy setting in such a way that a low regret on the sequence of points queried can be guaranteed. The new algorithm crucially relies on a *center-point device* that demonstrates the key differences between a regret minimization and an optimization guarantee. Our algorithm has the optimal $\mathcal{O}(\sqrt{T})$ scaling of regret up to logarithmic factors. However, our regret guarantee has a rather large dimension dependence. As noted after Theorem 2, this is unsurprising since the algorithm of Nemirovski and Yudin [16] has a large dimension dependence even in a noiseless case. Random walk approaches [4] have been successful to improve the dimension scaling in the noiseless case, and investigating them for the noisy scenario is an interesting question for future research.

Appendix A. Properties of pyramid constructions. We outline some properties of the pyramid construction in this appendix. Recall that $\varphi = \arccos(c_2/d)$. For simplicity, we assume $d \ge 2$. In this case, $\cos(\varphi) = c_2/d$ and

$$\sin(\varphi) = \sqrt{1 - c_2^2/d^2} \ge \cos(\varphi).$$

Also recall that in epoch τ, the initial simplex is contained in $\mathbb{B}(r_\tau)$, where $r_\tau = R_\tau/(c_1 d^{3/2})$.

LEMMA 15. *Let Π_k be the kth pyramid constructed in any round of epoch τ.*

1. *The distance from the center of $\mathbb{B}(r_\tau)$ to the apex of Π_k is $r_\tau \sin^{k-1}(\varphi)$.*
2. *The distance from the apex of Π_k to any vertex of the base of Π_k is $r_\tau \sin^{k-1}(\varphi) \cos(\varphi)$.*
3. *The height of Π_k (distance of the apex from the base) is $r_\tau \sin^{k-1}(\varphi)\cos^2(\varphi)$.*

Proof. The proof is by induction on k. Let x_0 be the center of $\mathbb{B}(r_\tau)$, y_1 be the apex of Π_1, and z_1 be any vertex on the base of Π_1. By construction, $y_1 - z_1$ is perpendicular to $z_1 - x_0$, so we have $\|y_1 - x_0\| = r_\tau$, $\|y_1 - z_1\| = r_\tau\cos(\varphi)$, and $\|z_1 - x_0\| = r_\tau\cos(\varphi)$. Let p_1 be the projection of y_1 onto the base of Π_1. The triangle with vertices y_1, z_1, x_0 is similar to the triangle with vertices y_1, p_1, z_1. Therefore $\|y_1 - p_1\|$, the height of Π_1, is $r_\tau\cos^2(\varphi)$. This gives the base case of the induction (see Figure 13).

The inductive step follows by noting that the apex of Π_k is a vertex on the base of Π_{k-1}, and therefore the distances scale as claimed. ∎

LEMMA 16. *Let Π be any pyramid constructed in epoch τ with apex at distance $r_\Pi \ge r_\tau/d$ from the center of $\mathbb{B}(r_\tau)$. Let $\mathbb{B}_\Pi$ be the largest ball in Π centered at the center of mass c of Π.*

1. *$\mathbb{B}_\Pi$ has radius at least $r_\Pi\cos^2(\varphi)/(d+1) \ge r_\tau c_2^2/(2d^4)$.*
2. *Let $x \in \Pi$, and let $b \in \Pi$ be the point on the face of Π such that $c = \alpha x + (1-\alpha)b$ for some $0 < \alpha \le 1$. Then $(1-\alpha)/\alpha \le (d+1)d/c_2$.*

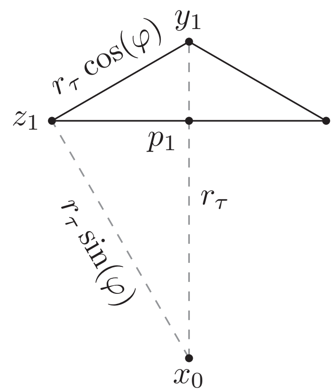

FIG. 13. *Construction of pyramids.*

Proof. Let h be the height of Π. By Lemma 15, $h = r_\Pi \cos^2(\varphi)$. The distance from c to the base of Π is

$$\frac{h}{d+1} = \frac{r_\Pi \cos^2(\varphi)}{d+1},$$

and the distance from c to any other face of Π is

$$\sin(\varphi)\left(1 - \frac{1}{d+1}\right) h = \sqrt{1-\cos^2(\varphi)}\left(1 - \frac{1}{d+1}\right) r_\Pi \cos^2(\varphi) \geq \frac{r_\Pi \cos^2(\varphi)}{2}.$$

(Here we have used $d \geq 2$ and $\cos(\varphi) \leq 1/d$.) Therefore $\mathbb{B}_\Pi$ has radius at least

$$\frac{r_\Pi \cos^2(\varphi)}{d+1} \geq \frac{r_\tau}{d} \cdot \frac{c_2^2/d^2}{d+1} = \frac{r_\tau c_2^2}{d^3(d+1)} \geq \frac{r_\tau c_2^2}{2d^4},$$

which proves the first claim.

For the second claim, note that $\alpha = \|b-c\|/(\|b-c\| + \|x-c\|)$; moreover, $\|b-c\|$ is at least the radius of $\mathbb{B}_\Pi$, and $\|x-c\|$ is at most the distance from c to any vertex of Π. By Lemma 15, the distance from c to a vertex on the base of Π is

$$\sqrt{\left(\frac{r_\Pi}{d+1}\cos^2(\varphi)\right)^2 + (r_\Pi \cos(\varphi)\sin(\varphi))^2} = \frac{r_\Pi \cos^2(\varphi)}{d+1}\sqrt{1 + \frac{(d+1)^2 \sin^2(\varphi)}{\cos^2(\varphi)}}$$

and the distance from c to the apex of Π is

$$\left(1 - \frac{1}{d+1}\right) h = \left(1 - \frac{1}{d+1}\right) r_\Pi \cos^2(\varphi) = \frac{d}{d+1} r_\Pi \cos^2(\varphi).$$

Therefore, by the first claim and Lemma 15,

$$\begin{aligned}\frac{1-\alpha}{\alpha} = \frac{\|x-c\|}{\|b-c\|} &\le \max\left\{\frac{\frac{dr_\Pi \cos^2(\varphi)}{d+1}}{\frac{r_\Pi \cos^2(\varphi)}{d+1}},\ \frac{\frac{r_\Pi \cos^2(\varphi)}{d+1}\sqrt{1+\frac{(d+1)^2\sin^2(\varphi)}{\cos^2(\varphi)}}}{\frac{r_\Pi \cos^2(\varphi)}{d+1}}\right\} \\ &= \max\left\{d,\ \sqrt{1+(d+1)^2\left(\frac{1}{\cos^2(\varphi)}-1\right)}\right\} \\ &\le \max\left\{d,\ \sqrt{\frac{(d+1)^2}{\cos^2(\varphi)}}\right\} \\ &= \max\left\{d,\ \frac{d+1}{\cos(\varphi)}\right\} \\ &= \max\left\{d,\ \frac{(d+1)d}{c_2}\right\} \\ &= \frac{(d+1)d}{c_2}. \qquad \square\end{aligned}$$

Acknowledgment. Part of this work was done while the first and third authors were at the University of Pennsylvania.

REFERENCES

[1] A. Agarwal, O. Dekel, and L. Xiao, *Optimal algorithms for online convex optimization with multi-point bandit feedback*, in Proceedings of COLT, 2010.

[2] R. Agrawal, *The continuum-armed bandit problem*, SIAM J. Control Optim., 33 (1995), pp. 1926–1951.

[3] P. Auer, R. Ortner, and C. Szepesvári, *Improved rates for the stochastic continuum-armed bandit problem*, in Proceedings of COLT, 2007, pp. 454–468.

[4] D. Bertsimas and S. Vempala, *Solving convex programs by random walks*, J. ACM, 51 (2004), pp. 540–556.

[5] S. Bubeck, R. Munos, G. Stolz, and C. Szepesvári, *$\mathcal{X}$-armed bandits*, J. Mach. Learn. Res., 12 (2011), pp. 1655–1695.

[6] V. V. Buldygin and Yu. V. Kozachenko, *Sub-Gaussian random variables*, Ukrainian Math. J., 32 (1980), pp. 483–489.

[7] A. R. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to Derivative-Free Optimization*, SIAM, Philadelphia, 2009.

[8] E. W. Cope, *Regret and convergence bounds for a class of continuum-armed bandit problems*, IEEE Trans. Automat. Control, 54 (2009), pp. 1243–1253.

[9] V. Dani, T. P. Hayes, and S. M. Kakade, *Stochastic linear optimization under bandit feedback*, in Proceedings of the 21st Annual Conference on Learning Theory (COLT), 2008.

[10] A. D. Flaxman, A. T. Kalai, and B. H. Mcmahan, *Online convex optimization in the bandit setting: Gradient descent without a gradient*, in Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms, 2005, pp. 385–394.

[11] D. Goldfarb and M. J. Todd, *Modifications and implementation of the ellipsoid algorithm for linear programming*, Math. Program., 23 (1982), pp. 1–19.

[12] J. Kiefer and J. Wolfowitz, *Stochastic estimation of the maximum of a regression function*, Ann. Math. Statist., 23 (1952), pp. 462–466.

[13] R. Kleinberg, *Nearly tight bounds for the continuum-armed bandit problem*, Adv. Neural Inf. Process. Syst., 18 (2005).

[14] R. Kleinberg, A. Slivkins, and E. Upfal, *Multi-armed bandits in metric spaces*, in Proceedings of the 40th Annual ACM Symposium on Theory of Computing, 2008, pp. 681–690.

[15] L. Lovász, *Geometric algorithms and algorithmic geometry*, in Proceedings of International Congress of Mathematicians, 1990, pp. 139–154.

[16] A. Nemirovski and D. Yudin, *Problem Complexity and Method Efficiency in Optimization*, Wiley, New York, 1983.
[17] Y. Nesterov, *Random Gradient-Free Minimization of Convex Functions*, Technical report 2011/1, Center for Operations Research and Econometrics, Université catholique de Louvain, 2011.
[18] M. Raginsky and A. Rakhlin, *Information-based complexity, feedback and dynamics in convex programming*, IEEE Trans. Inform. Theory, 57 (2011), pp. 7036–7056.
[19] N. Srinivas, A. Krause, S.M. Kakade, and M. Seeger, *Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design*, arXiv:0912.3995, 2009.
[20] J. Y. Yu and S. Mannor, *Unimodal bandits*, in Proceedings of ICML, 2011.