Stochastic Simultaneous Optimistic Optimization

Michal Valko MICHAL.VALKO@INRIA.FR
INRIA Lille - Nord Europe, SequeL team, 40 avenue Halley 59650, Villeneuve d'Ascq, France

Alexandra Carpentier A.CARPENTIER@STATSLAB.CAM.AC.UK
Statistical Laboratory, CMS, Wilberforce Road, CB3 0WB, University of Cambridge, United Kingdom

Rémi Munos REMI.MUNOS@INRIA.FR
INRIA Lille - Nord Europe, SequeL team, 40 avenue Halley 59650, Villeneuve d'Ascq, France

Abstract

We study the problem of global maximization of a function f given a finite number of evaluations perturbed by noise. We consider a very weak assumption on the function, namely that it is locally smooth (in some precise sense) with respect to some semi-metric, around one of its global maxima. Compared to previous works on bandits in general spaces (Kleinberg et al., 2008; Bubeck et al., 2011a) our algorithm does not require the knowledge of this semi-metric. Our algorithm, `StoSOO`, follows an optimistic strategy to iteratively construct upper confidence bounds over the hierarchical partitions of the function domain to decide which point to sample next. A finite-time analysis of `StoSOO` shows that it performs almost as well as the best specifically-tuned algorithms even though the local smoothness of the function is not known.

1. Introduction

We consider a function maximization problem of an unknown function $f : \mathcal{X} \to \mathbb{R}$. We assume that every function evaluation is costly, and therefore we are interested in optimizing the function given a finite budget of n evaluations. Moreover, the evaluations are perturbed by noise, i.e., the evaluation of f at a point $x_t \in \mathcal{X}$ returns a noisy evaluation r_t, assumed to be independent from the previous ones, such that:

$$\mathbb{E}[r_t | x_t] = f(x_t). \tag{1}$$

One motivation for this setting is a measurement error when dealing with a stochastic environment. Another example is the optimization of some parametric policy operating in a stochastic system.

We assume that there exists at least one global maximizer $x^* \in \mathcal{X}$ of f, i.e. $f(x^*) = \sup_{x \in \mathcal{X}} f(x)$. We aim for an algorithm which sequentially evaluates f at points $x_1, x_2, \ldots, x_n$ in the search space $\mathcal{X}$ to find a good approximation to a global maximum. After n function evaluations the algorithm outputs a point $x(n)$ and its performance is measured with the loss:

$$R_n = \sup_{x \in \mathcal{X}} (f(x)) - f(x(n)) \tag{2}$$

Our definition of loss is very related to the *simple regret* in multi-armed bandits (Bubeck et al., 2009). Many algorithms have been developed for this general optimization problem. However, a lot of them require some assumption on the *global* smoothness of f, most typically, they assume a global *Lipschitz* property (Pintér, 1995; Strongin & Sergeyev, 2000; Hansen & Walster, 2004; Kearfott, 1996; Neumaier, 2008). There has been also an interest in designing sample-efficient strategies, only requiring *local* smoothness around (one) of the global maxima (Kleinberg et al., 2008; Bubeck et al., 2011a; Munos, 2011). However, these approaches still assume the *knowledge* of this smoothness, i.e., the metric under which the function is smooth, which may not be available to the optimizer.

Recently, Munos (2011) proposed the SOO algorithm for *deterministic* optimization, that assumes that f is locally smooth with respect to some semi-metric ℓ, but that this semi-metric *does not need to be known* to the algorithm. SOO extends the DIRECT algorithm (Jones et al., 1993) and other Lipschitz optimization without the knowledge of the Lipschitz constant (Bubeck et al., 2011b; Slivkins, 2011) to the case of any possible semi-metric by *simultaneously* considering the subspaces that can contain the optimum.

Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28.

In this paper, we provide an extension of SOO to the case of noisy evaluations, which we call Stochastic SOO, or StoSOO. One major difference from SOO is that we cannot base our exploration strategy only on a single evaluation per cell since we are dealing with stochastic functions. Another difference is that we cannot simply return the highest evaluated point we encountered as $x(n)$ since it is subject to noise. Our analysis shows that in a large class of functions (precisely defined in Section 5), the loss of StoSOO is $\tilde{O}(n^{-1/2})$, which is of same order as the loss of HOO (Bubeck et al., 2011a) or Zooming algorithm (Kleinberg et al., 2008) when using the best possible metric.

2. Background

Optimistic optimization refers to approaches that implement the *optimism in the face of uncertainty* principle. This principle became popular in the multi-armed bandit problem (Auer et al., 2002) and was later extended to the tree search (Kocsis & Szepesvári, 2006; Coquelin & Munos, 2007) where it is referred to as *hierarchical bandit approach*. The reason is that a complex problem such as global optimization of the space $\mathcal{X}$ is treated as a hierarchy of simple bandit problems. It is therefore an example of Monte Carlo tree search which was shown to be empirically successful for instance in computer Go (Gelly et al., 2012).

Optimistic optimization was also used in many other domains, such as planning (Hren & Munos, 2008; Bubeck et al., 2011a) or Gaussian process optimization (Srinivas et al., 2010). This paper applies optimistic approach to a global *black-box* function optimization. Table 1 displays representative approaches for this setting. The case when the smoothness of the function f is known, means that the function is either (globally) Lipschitz, weakly Lipschitz or locally Lipschitz around the optimum. There are numerous algorithms for this setting, the most related to our work are DOO (Munos, 2011) for the deterministic case and Zooming (Kleinberg et al., 2008) or HOO (Bubeck et al., 2011a) for the stochastic one[1]. This setting has been also considered in a Bayesian framework, in particular the *expected-improvement* strategy (Osborne, 2010) which was theoretically analyzed when the assumption of smoothness is data-driven (Bull, 2011).

One of the disadvantages of these algorithms is that however strong or mild are the assumptions on f, the quantities that express them (i.e. a prior, a Lipschitz constant, or a semi-metric in DOO) need to be *known* to the algorithm. On the other hand, for the case of deterministic functions there exist approaches that do not require this knowledge, such as DIRECT or SOO.

[1]Note that the loss (2) considered here is different but related to the usual cumulative regret defined in the bandit setting, see e.g. (Bubeck et al., 2009).

Table 1. Hierarchical optimistic optimization algorithms

	deterministic	**stochastic**
known smoothness	DOO	Zooming or HOO
unknown smoothness	DIRECT or SOO	StoSOO this paper

However, neither DIRECT nor SOO can deal with stochastic functions. Therefore, we extend the SOO algorithm to the stochastic setting and provide a finite-time analysis of its performance.

3. Algorithm

StoSOO is a tree-search based algorithm that iteratively constructs finer and finer partition of the search space $\mathcal{X}$. The partitions are represented as nodes of a K-ary tree $\mathcal{T}$ and the nodes are organized by their depths $h \geq 0$, with $h = 0$ being the root node, and indexed by $1 \leq i \leq K^h$. We denote $\circ[h, i]$, the i-th node at depth h. Each of the nodes $\circ[h, i]$ corresponds to a cell $\mathcal{X}_{h,i} \subseteq \mathcal{X}$ in the partitioning, i.e., to a subset of $\mathcal{X}$ with an associated representative point $x_{h,i} \in \mathcal{X}_{h,i}$.

3.1. Assumptions

We now state our main assumption, which is also used in SOO (Munos, 2011). The first part of the assumption is about the existence of a semi-metric ℓ such that the function f is locally smooth with respect to it. We stress that although it quantifies the smoothness of f, it **only requires the existence** of ℓ and *not* the *knowledge* of it. For illustrative examples and discussion on this part we refer the reader to (Munos, 2011). The second part is about the structure of the hierarchical partitioning with respect to ℓ. This partitioning is fixed and given to the algorithm as a parameter.

Assumption There exists a semi-metric $\ell : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$ (i.e. for $x, y \in \mathcal{X}$, we have $\ell(x, y) = \ell(y, x)$ and $\ell(x, y) = 0$ if and only if $x = y$) such that:

A1 **(local smoothness of f):** For all $x \in \mathcal{X}$:

$$f(x^*) - f(x) \leq \ell(x, x^*). \tag{3}$$

A2 **(bounded diameters and well-shaped cells):** There exists a decreasing sequence $w(h) > 0$, such

that for any depth $h \geq 0$ and for any cell $\mathcal{X}_{h,i}$ of depth h, we have $\sup_{x \in X_{h,i}} \ell(x_{h,i}, x) \leq w(h)$. Moreover, there exists $\nu > 0$ such that for any depth $h \geq 0$, any cell $\mathcal{X}_{h,i}$ contains a ℓ-ball of radius $\nu w(h)$ centered in $x_{h,i}$.

Assumption A1 guarantees that f does not decrease too fast around one global optimum x^*. This can be thought of as a one-sided local Lipschitz assumption. Note that although we require that (3) is satisfied for all $x \in \mathcal{X}$, this assumption essentially sets constraints to the function f locally around x^*, since when x is such that $\ell(x, x^*) > \sup f - \inf f$, then the assumption is automatically satisfied. Thus when this property holds, we say that f **is locally smooth with respect to ℓ around its maximum**.

Assumption A2 assures the regularity of the partitioning, in particular that the size of the cells decreases with their depths and that their shape is not skewed in some dimensions.

3.2. Stochastic SOO

Algorithm 1 displays the pseudo-code of the `StoSOO` algorithm. The algorithm operates in the traversals of the tree starting from the root down to the current depth($\mathcal{T}$), that is upper bounded by $h_{\max}$, a parameter of the algorithm. During each traversal (a whole pass of the "for" cycle) `StoSOO` selects a set of promising nodes, at most one per depth h. These nodes are then either *evaluated* or *expanded*.

Evaluating a node at time t means sampling the function in the representative point $x_{h,i}$ of the cell $\mathcal{X}_{h,i}$ and observing the evaluation r_t according to (1). Expanding a node $\circ[h,i]$, means splitting its corresponding cell into its K sub-cells corresponding to the children:

$$\{\circ[h+1, i_1], \circ[h+1, i_2], \ldots, \circ[h+1, i_K]\}.$$

We denote by $\mathcal{L}$ the set of leaves in $\mathcal{T}$, i.e. the nodes with no children. At any time, only the leaves are eligible for an evaluation or expansion and we never expand the leaves beyond depth $h_{\max}$. If the function f were deterministic, such as in SOO (Munos, 2011), we would expand (simultaneously) any leaf $\circ[h,i]$ whose value $f(x_{h,i})$ is the largest among all leaves of the same or a lower depth. The reason for this choice is that by Assumption A1 all such nodes may contain x^*. Unfortunately, we do not receive $f(x_{h,i})$, but only a noisy estimate r_t. Therefore, the main algorithmic idea of `StoSOO` is to evaluate the leaves several times in order to build a confident estimate of $f(x_{h,i})$. For this purpose, let us define $\hat{\mu}_{h,i}(t) = \frac{1}{T_{h,i}(t)} \sum_{s=1}^{t} r_s \mathbb{1}\{x_s \in \mathcal{X}_{h,i}\}$ the empirical average of rewards obtained at state $x_{h,i}$ at time t, where $T_{h,i}(t)$ is the number of times that $\circ[h,i]$ has been sampled up to time t.

Algorithm 1 `StoSOO`

Stochastic Simultaneous Optimistic Optimization

Parameters: number of function evaluations n, maximum number of evaluations per node $k > 0$, maximum depth $h_{\max}$, and $\delta > 0$.
Initialization:
  $\mathcal{T} \leftarrow \{\circ[0,0]\}$ {root node}
  $t \leftarrow 0$ {number of evaluations}
while $t \leq n$ **do**
  $b_{\max} \leftarrow -\infty$
  for $h = 0$ to $\min(\text{depth}(\mathcal{T}), h_{\max})$ **do**
    if $t \leq n$ **then**
      For each leaf $\circ[h,j] \in \mathcal{L}$, compute its b-value:
        $b_{h,j}(t) = \hat{\mu}_{h,j}(t) + \sqrt{\log(nk/\delta)/(2T_{h,j}(t))}$
      Among leaves $\circ[h,j] \in \mathcal{L}_t$ at depth h, select

$$\circ[h,i] \in \arg\max_{\circ[h,j] \in \mathcal{L}} b_{h,j}(t)$$

      if $b_{h,i}(t) \geq b_{\max}$ **then**
        if $T_{h,i}(t) < k$ **then**
          Evaluate (sample) state $x_t = x_{h,i}$.
          Collect reward r_t (s.t. $\mathbb{E}[r_t | x_t] = f(x_t)$).
          $t \leftarrow t + 1$
        else {i.e. $T_{h,i}(t) \geq k$, expand this node}
          Add the K children of $\circ[h,i]$ to $\mathcal{T}$
          $b_{\max} \leftarrow b_{h,i}(t)$
        end if
      end if
    end if
  end for
end while
Output: The representative point with the highest $\hat{\mu}_{h,j}(n)$ among the deepest expanded nodes:

$$x(n) = \arg\max_{x_{h,j}} \hat{\mu}_{h,j}(n) \text{ s.t. } h = \text{depth}(\mathcal{T} \setminus \mathcal{L}).$$

`StoSOO` builds an accurate estimate of $f(x_{h,i})$ before $\circ[h,i]$ is expanded. To achieve this, we define an upper confidence bound (or a b-value) for each node $\circ[h,i]$ as:

$$b_{h,i}(t) \stackrel{\text{def}}{=} \hat{\mu}_{h,i}(t) + \sqrt{\frac{\log(nk/\delta)}{2T_{h,i}(t)}}, \tag{4}$$

where δ is the confidence parameter. In the case of $T_{h,i}(t) = 0$, we let $b_{h,i}(t) = \infty$. We refer to $\sqrt{\log(nk/\delta)/2T_{h,i}(t)}$ as to the *width* of the estimate. Now instead of selecting the promising nodes according to their values $f(x_{h,i})$ we select them according to their b-values $b_{h,i}$.

Our algorithm is *optimistic* since it considers such leaves for the selection whose b-value is *maximal* among leaves at depth h or lower depths, since those leaves are likely to contain the optimum x^* at time t, given the observed samples and Assumption A1 on f.

The important question is now how many times should we evaluate the node before we decide to expand it. Again, if we knew the semi-metric ℓ we would be able to calculate the appropriate count for each depth h. Since we do not know it, we instead evaluate each node a fixed number of k times before its expansion. We address the setting of k, $h_{\max}$, and δ in Sections 4 and 5. Our analysis shows that under appropriate assumptions on f (discussed in Section 5) we can bound the expected regret as $\mathbb{E}[R_n] = O\left(\log^2(n)/\sqrt{n}\right)$ by setting $k = n/\log^3(n)$, $h_{\max} = \sqrt{n/k}$, and $\delta = 1/\sqrt{n}$.

In the algorithm, we keep track of the number of evaluations t in order to finish when it reaches n, the maximum number of evaluations, i.e., the budget. Since we are facing a stochastic setting, we cannot simply output the value that received the highest reward during n evaluations, as it is the case in most of the deterministic approaches. Instead, we return the representative point $x_{h,j}$ of the node with the highest estimate $\hat{\mu}_{h,j}(n)$ among the deepest expanded nodes, i.e., such that $h = \text{depth}(\mathcal{T} \setminus \mathcal{L})$.

4. Analysis

In this section we analyze the performance of StoSOO and upper bound the loss (2) as a function of the number of evaluations. We assume that the rewards are bounded[2] by $|r_t| \leq 1$ for any t. In order to derive a loss bound we define a measure of the quantity of near-optimal states, called *near-optimality dimension*. This measure is closely related to similar measures (Kleinberg et al., 2008; Bubeck et al., 2008). For any $\varepsilon > 0$, let us write the set of ε-optimal states as:

$$\mathcal{X}_\varepsilon \stackrel{\text{def}}{=} \{x \in \mathcal{X}, f(x) \geq f^* - \varepsilon\}.$$

Definition 1. *The ν-near-optimality dimension is the smallest $d \geq 0$ such that there exists $C > 0$ such that for any $\varepsilon > 0$, the maximum number of disjoint ℓ-balls of radius $\nu\varepsilon$ and center in $\mathcal{X}_\varepsilon$ is less than $C\varepsilon^{-d}$.*

StoSOO maintains the upper confidence bounds (b-values) for each cell in order to decide which cell to sample or expand. We start by quantifying the probability that all the average estimates $\hat{\mu}_{h,j}(t)$ are at any time t within those $b_{h,j}(t)$-values. For this purpose we define the event in which this occurs and then show that this event happens with high probability.

Lemma 1. *Let ξ be the event under which all average estimates are within their widths:*

$$\xi \stackrel{\text{def}}{=} \Big\{\forall h, i, t \text{ s.t. } h \geq 0, 1 \leq i < K^h, 1 \leq t \leq n, \text{and}$$
$$T_{h,j}(t) > 0 : \left|\hat{\mu}_{h,j}(t) - f(x_{h,j})\right| \leq \sqrt{\frac{\log(nk/\delta)}{2T_{h,j}(t)}}\Big\},$$

then $\mathbb{P}(\xi) \geq 1 - \delta$.

Proof. Let m denote the (random) number of different nodes sampled by the algorithm up to time n. Let τ_i^1 be the first time when the i-th new node $\circ[H_i, J_i]$ is sampled, i.e., at time $\tau_i^1 - 1$ there are only $i-1$ different nodes that have been sampled whereas at time τ_i^1, the i-th new node $\circ[H_i, J_i]$ is sampled for the first time. Let τ_i^s, for $1 \leq s \leq T_{H_i,J_i}(n)$, be the time when the node $\circ[H_i, J_i]$ is sampled for the s-th time. Moreover, we denote $Y_i^s = r_{\tau_i^s} - f(x_{H_i,J_i})$. Using this notation, we rewrite ξ as:

$$\xi = \Bigg\{\forall i, u \text{ s.t. } , 1 \leq i \leq m, 1 \leq u \leq T_{H_i,J_i}(n),$$
$$\left|\frac{1}{u}\sum_{s=1}^{u} Y_s^i\right| \leq \sqrt{\frac{\log(nk/\delta)}{2u}}\Bigg\}. \tag{5}$$

Now, for any i and u, the $(Y_i^s)_{1\leq s\leq u}$ are i.i.d. from some distribution ν_{H_i,J_i}. The node $\circ[H_i, J_i]$ is random and depends on the past samples (before time τ_i^1) but the $(Y_i^s)_s$ are conditionally independent given this node and consequently:

$$\mathbb{P}\left(\left|\frac{1}{u}\sum_{s=1}^{u} Y_s^i\right| \leq \sqrt{\frac{\log(nk/\delta)}{2u}}\right) =$$
$$= \mathbb{E}_{\circ[H_i,J_i]}\,\mathbb{P}\left(\left|\frac{1}{u}\sum_{s=1}^{u} Y_s^i\right| \leq \sqrt{\frac{\log(nk/\delta)}{2u}} \;\middle|\; \circ[H_i, J_i]\right)$$
$$\geq 1 - \frac{\delta}{nk},$$

using Chernoff-Hoeffding's inequality. We finish the proof by taking a union bound over all values of $1 \leq i \leq n$ and $1 \leq u \leq k$. □

Lemma 1 shows that when the leaf is expanded then with high probability the mean estimate $\hat{\mu}_{h,j}(t)$ is very close to its true value. Specifically, when the node is expanded then with probability $1 - \delta$ uniformly for all h, j, and t, we have that:

$$|\hat{\mu}_{h,j}(t) - f(x_{h,j})| \leq \varepsilon, \tag{6}$$

[2] The analysis can be easily extended to the case when the noise is sub-Gaussian.

where $\varepsilon = \sqrt{\log(nk/\delta)/2k}$. We use this lemma to show that the expanded nodes are with high probability close to optimal.

Definition 2. *Let the expansion set at depth h be the set of all nodes that could be potentially expanded before the optimal node at depth h is expanded:*[3]

$$I_h^\varepsilon \stackrel{\text{def}}{=} \{\text{nodes } \circ[h,i] \text{ such that } f(x_{h,i})+w(h)+2\varepsilon \geq f^*\}.$$

Recall that even though this definition uses $w(h)$ that depends on the unknown metric ℓ, the `StoSOO` algorithm does not need to know it. Now, let us denote h_t^* the deepest depth of the expanded node at time t, that contains the optimum x^*. Notice that in general the algorithm may have at time t also expanded some (suboptimal) nodes in the deeper depths. In the following, we show that they are not too many of these. Specifically, for each depth h, we lower bound the number of evaluations after which the h_t^* needs to be at least h.

Lemma 2. *Let depth $h \in \{0, h_{\max}\}$ be any depth and:*

$$t_h \stackrel{\text{def}}{=} (k+1)h_{\max}(|I_0^\varepsilon| + |I_1^\varepsilon| + \cdots + |I_h^\varepsilon|).$$

After we evaluated at least $t \geq t_h$ nodes, then in the event ξ, the depth h_t^* of the deepest node in the optimal branch is at least h, i.e., $h_t^* \geq h$.

Proof. By induction on h. For $h = 0$, the lemma holds trivially since $h_t^* \geq 0$. For the induction step, let us assume that the lemma holds for all $h \in \{0, \ldots, h'\}$, where $h' < h_{\max}$ and we are to show it holds for $h'+1$ as well. Assume we have already evaluated $t_{h'+1}$ nodes, i.e. that we are at time $t \geq t_{h'+1}$. Since t_h is increasing in h, we have also evaluated $t_{h'}$ nodes and $h_t^* \geq h'$ from the induction step. That means that the optimal branch is expanded at least up to the depth h'. Now consider any node $\circ[h'+1, i]$ at depth $h'+1$, that was expanded. If it was expanded before the optimal node $\circ[h'+1, i^*]$ at depth $h'+1$ was expanded, then $b_{h_t^*+1,i}(t) \geq b_{h_t^*+1,i^*}(t)$. According to Lemma 1, the average estimates $\hat{\mu}_{h,j}(t)$ are at most ε away from their true values, with ε defined in (6). Therefore in the event ξ, the true values of the expanded and the optimal node are at most 2ε apart:

$$f(x_{h_t^*+1,i}) \geq f(x_{h_t^*+1,i^*}) - 2\varepsilon. \tag{7}$$

Since the node $\circ[h_t^*+1, i^*]$ contains the optimum x^*, then by Assumptions A1-2, we get:

$$f(x_{h_t^*+1,i^*}) + w(h+1) \geq f(x_{h_t^*+1,i^*}) + \ell(x_{h_t^*+1,i^*}, x^*) \geq f^*.$$

[3] The reason for such definition will become apparent in the proof of Lemma 2.

Combining this with (7), we obtain that:

$$f(x_{h_t^*+1,i}) \geq f(x_{h_t^*+1,i^*}) - 2\varepsilon \geq f^* - \big[w(h_t^*+1) + 2\varepsilon\big].$$

This means that all the nodes $\circ[h'+1, i]$ expanded before $\circ[h'+1, i^*]$ are $[w(h_t^*+1)+2\varepsilon]$-optimal. By Definition 2, there are exactly $|I_{h'+1}^\varepsilon|$ such nodes. Each traversal of the tree in the `StoSOO` algorithm selects one of these nodes for evaluation. Since k evaluations are required before the expansion, after $(k+1)|I_{h'+1}^\varepsilon|$ traversals, $\circ[h'+1, i^*]$ must have been expanded. To guarantee this many traversals, we need $(k+1)h_{\max}|I_{h'+1}^\varepsilon|$ evaluations after t'_h previous evaluations. This is equal to $t_{h'+1}$ and thus $h_t^* \geq h'+1$. □

Lemma 2 bounds the number of needed evaluations in the terms of the expansion set sizes to assure that the optimal node was expanded. Naturally, we would like to know, how big these expansion sets can be. The following lemma upper bounds the size of expansion sets up to depth where $w(h)$ is of the order of ε. For this purpose, we define h_ε as:

$$h_\varepsilon = \arg\min\{h \in \mathbb{N} : w(h+1) < \varepsilon\}. \tag{8}$$

Lemma 3. *Let d be a $\nu/3$-near-optimality dimension and C the related constant. Then for each $h \leq h_\varepsilon$, the cardinality of the expansion set at depth h is in the event ξ bounded as:*

$$|I_h^\varepsilon| \leq C\left(w(h) + 2\varepsilon\right)^{-d}.$$

Proof. By contradiction. Assume that for some $h \leq h_\varepsilon$, $|I_h^\varepsilon| > C\left(w(h)+2\varepsilon\right)^{-d}$. By definition of $|I_h^\varepsilon|$, each representative point $x_{h,i}$ of the node $\circ[h,i]$ is $[w(h)+2\varepsilon]$-optimal. By Assumption A2, each cell associated with the node $\circ[h,i]$ at depth h contains a ball of radius $\nu w(h) = \frac{\nu}{3} \cdot 3w(h) \geq \frac{\nu}{3}(w(h)+2\varepsilon)$ with the representative point $x_{h,i}$, because for $h \leq h_\varepsilon$, we have that $\varepsilon \leq w(h)$ by (8). Since the cells are disjoint, we have a contradiction with $\nu/3$-near-optimality dimension being d. □

We now link the depth of the tree after n iterations with the loss as defined in (2).

Theorem 1. *Assume that Assumptions A1-2 hold. Let d be the $\nu/3$-near-optimality dimension and C be the corresponding constant. Then the loss of* `StoSOO` *run with parameters k, $h_{\max}$, and $\delta > 0$, after n iterations is bounded, with probability $1-\delta$, as:*

$$R_n \leq 2\varepsilon + w\left(\min\left(h(n) - 1, h_\varepsilon, h_{\max}\right)\right)$$

where $\varepsilon = \sqrt{\log(nk/\delta)/(2k)}$ and $h(n)$ is the smallest $h \in \mathbb{N}$, such that:

$$C(k+1)h_{\max}\sum_{l=0}^{h}\left(w(l)+2\varepsilon\right)^{-d} \geq n.$$

Proof. Let us first consider the case when $h(n) - 1 \leq h_\varepsilon$. Then we can use Lemma 3 to show that:

$$n > C(k+1)h_{\max} \sum_{l=0}^{h(n)-1} \left(w\left(l\right) + 2\varepsilon\right)^{-d} \geq (k+1)h_{\max} \sum_{l=0}^{h(n)-1} |I_l^\varepsilon| = t_{h(n)-1} \quad (9)$$

If $h(n) - 1 \leq h_{\max}$ then by Lemma 2, $h_n^* \geq h(n) - 1$. If, however, $h(n) - 1 > h_{\max}$, then by (9) the algorithm has expanded all potentially optimal nodes on the level $h_{\max}$ and therefore $h_n^* \geq h_{\max}$. Nonetheless the algorithm does not go beyond $h_{\max}$, so necessarily $h_n^* = h_{\max}$. Hence, in the case when $h(n) - 1 \leq h_\varepsilon$, $h_n^* \geq \min\{h(n) - 1, h_{\max}\}$. Now consider the opposite case, i.e., when $h(n) - 1 \geq h_\varepsilon + 1$. We can now use Lemma 3, but only up to depth h_ε, to get that $n > t_{h_\varepsilon}$. Similarly to the previous case, we deduce that $h_n^* \geq \min\{h_\varepsilon, h_{\max}\}$. Altogether, $h_n^* \geq \min\{h(n)-1, h_\varepsilon, h_{\max}\}$. Let $\circ[h, j]$ be the deepest node that has been expanded after n evaluations. We know that $h \geq h_n^*$. Let also $\circ[h_n^*, i^*]$ be the optimal node at the depth h_n^*. As $\circ[h, j]$ was expanded, the true value of its representative point and the representative point of $\circ[h_n^*, i^*]$ is in the event ξ at most 2ε away and therefore we conclude that:

$$\begin{aligned} f(x_{h,j}) &\geq f(x_{h_n^*, i^*}) - 2\varepsilon \geq f^* - [w(h_n^*) + 2\varepsilon] \\ &\geq f^* - [w(\min\{h(n) - 1, h_\varepsilon, h_{\max}\}) + 2\varepsilon]. \end{aligned}$$

□

5. The important case $d = 0$

We now deduce the following corollaries for the case when the near-optimality dimension $d = 0$ and the diameters $w(h)$ are exponentially decreasing. We postpone the discussion about this important case $d = 0$ to Section 5.1.

Corollary 1. *Assume that the diameters of the cells decrease exponentially fast, i.e., $w(h) = c\gamma^h$ for some $c > 0$ and $\gamma < 1$. Assume that the $\nu/3$-near-optimality dimension is $d = 0$ and let C be the corresponding constant. Then the expected loss of* `StoSOO` *run with parameters k, $h_{\max} = \sqrt{n/k}$, and $\delta > 0$, is bounded as:*

$$\mathbb{E}[R_n] \leq (2 + 1/\gamma)\varepsilon + c\gamma^{\sqrt{n/k}\min\{0.5/C, 1\}-2} + 2\delta. \quad (10)$$

Proof. When $d = 0$, then $\left[w(l) + 2\varepsilon\right]^{-d} = 1$ and by definition of $h(n)$, we have that $n \leq C(k + 1)h_{\max} \sum_{l=0}^{h(n)} \left[w(l) + 2\varepsilon\right]^{-d} = C(k+1)h_{\max}(h(n)+1)$, which implies $h(n) \geq n/(C(k+1)h_{\max}) - 1$. Intuitively, the deeper is the node we return, the lower regret we can incur. This suggests the choice of $h_{\max} = \sqrt{n/k}$, in which case we get $h(n) \geq \sqrt{n}/(2C\sqrt{k}) - 1$, since $k \geq 1$. Moreover, since $w(h) = c\gamma^h$, then by definition of h_ε we have that:

$$w(h_\varepsilon) = c\gamma^{h_\varepsilon + 1}/\gamma = w(h_\varepsilon + 1)/\gamma < \varepsilon/\gamma.$$

By Theorem 1, we have that in the event ξ, the regret of `StoSOO` is at most:

$$\begin{aligned} R_n &\leq 2\varepsilon + w(\min\{h(n) - 1, h_\varepsilon, h_{\max}\}) \\ &\leq 2\varepsilon + w(h_\varepsilon) + w(\min\{h(n) - 1, h_{\max}\}) \\ &\leq (2 + 1/\gamma)\varepsilon + c\gamma^{\sqrt{n/k}\min\{0.5/C, 1\}-2} \end{aligned}$$

We obtain the upper bound on the expected loss (10), by considering that by Lemma 1, ξ holds with probability $1 - \delta$ and $|r_t| \leq 1$. □

Corollary 2. *For the choice $k = n/\log^3(n)$ and $\delta = 1/\sqrt{n}$, we have:*

$$\mathbb{E}[R_n] = O\Big(\frac{\log^2(n)}{\sqrt{n}}\Big).$$

This result shows that, surprisingly, `StoSOO` achieves the same rate $\tilde{O}(n^{-1/2})$, up to a logarithmic factor, as the HOO algorithm run with the best possible metric, although `StoSOO` does not requires the knowledge of it.

Proof. Setting $k = n/\log^3(n)$ and $\delta = 1/\sqrt{n}$ we can upper bound ε in (10) which was defined in (6) as:

$$\varepsilon = \sqrt{\frac{\log(nk/\delta)}{2k}} = \sqrt{\frac{\log(nk\sqrt{n})\log^3(n)}{2n}} \leq \sqrt{\frac{5}{4}}\frac{\log^2(n)}{\sqrt{n}}.$$

Now for n bigger than a quantity exponential in $C/\log(1/\gamma)$, the second term in (10) becomes negligible and the upper bound for this choice follows. □

5.1. Some intuition about the case $d = 0$

We have seen that the near-optimality dimension d is a property of both the function and the semi-metric ℓ. Since `StoSOO` does not require the knowledge of the semi-metric ℓ (it is only used in the analysis), one can choose the best possible semi-metric ℓ, **possibly according to the function f itself**, in order to have the lowest possible value of d. The case $d = 0$ thus corresponds to the following assumption on f: there exists a semi-metric ℓ such that: **1)** f is locally smooth w.r.t. ℓ around a global optimum x^* (i.e. such that (3) holds) **2)** the diameters of the cells (measured with ℓ) decrease exponentially fast, and **3)** there exists $C > 0$ such that for any $\varepsilon > 0$, the maximal number of disjoint ℓ-balls of radius $\nu\varepsilon/3$ centered in $\mathcal{X}_\varepsilon$ is less than C (i.e. the near-optimality dimension d is 0).

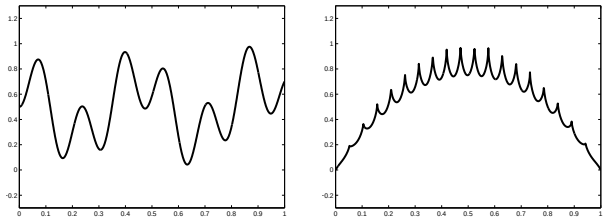

Figure 1. Functions with $d = 0$. **Left:** *Two-sine product* function $f_1(x) = \frac{1}{2}(\sin(13x)\cdot\sin(27x)) + 0.5$. **Right:** *Garland* function: $f_2(x) = 4x(1-x)\cdot(\frac{3}{4} + \frac{1}{4}(1 - \sqrt{|\sin(60x)|}))$.

5.2. Examples

Let us consider the case of functions f defined on $[0,1]^D$ that are locally equivalent to a polynomial of degree α around their maximum, i.e., $f(x) - f(x^*) = \Theta(\|x - x^*\|^\alpha)$ for some $\alpha > 0$, where $\|\cdot\|$ is any norm. The choice of semi-metric $\ell(x,y) = \|x-y\|^\alpha$ implies that the near-optimality dimension $d = 0$. This covers already a large class of functions (such as the functions plotted in Figure 1: the *two-sine product* function for which $\alpha = 2$ and the non-Lipschitz *garland* function for which $\alpha = 1/2$).

More generally, we consider a finite dimensional and bounded space, i.e., such that $\mathcal{X}$ can be packed by $C_{\mathcal{X}}\varepsilon^{-D}$ ℓ-balls with radius ε (e.g., Euclidean space $[0,1]^D$) and such that $\mathcal{X}$ has a finite doubling constant (defined as minimum value q such that every ball in $\mathcal{X}$ can be packed by at most q balls in $\mathcal{X}$ of half the radius). Let a function in such space have upper- and lower envelope around x^* of the same order (Figure 2), i.e., there exists constants $c \in (0,1)$, and $\eta > 0$, such that for all $x \in \mathcal{X}$:

$$\min(\eta, c\ell(x,x^*)) \le f(x^*) - f(x) \le \ell(x,x^*). \quad (11)$$

We show that all such functions have a near-optimality dimension $d = 0$ according to Definition 1, (where $\nu = 1$ for simplicity), which means that for all $\varepsilon > 0$, the packing number of $\mathcal{X}_\varepsilon$ is upper-bounded by a constant.

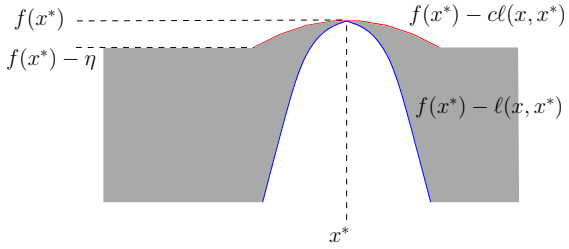

Figure 2. Any function satisfying (11) lies in the gray area and possesses a lower- and upper-envelopes that are of same order around x^*.

In the case when $\varepsilon < \eta$, due the upper envelope we have that: $\mathcal{X}_\varepsilon \subset \{x : c\ell(x,x^*) \le \varepsilon\}$, which corresponds to an ℓ-ball centered in x^* with radius ε/c. This ball can be packed by no more than a constant number of C' ℓ-balls of radius ε. C' is necessarily finite because the doubling constant q is finite. For example, if $\ell(x,y) = \|x-y\|_\infty$, then $C' = (1/c)^D$.

In the opposite case when $\varepsilon \ge \eta$, the radius of disjoint ℓ-balls that could possibly pack $\mathcal{X}_\varepsilon$ is at least η. Noting that $\mathcal{X}_\varepsilon \subset \mathcal{X}$, we can upper bound the packing number of the whole space $\mathcal{X}$, by a constant $C_{\mathcal{X}}(\eta)^{-D}$ that is independent of ε. Finally, defining $C = \max\{C', C_{\mathcal{X}}(\eta)^{-D}\}$ we have that for all ε, the maximum number of disjoint ℓ-balls of radius ε and center in $\mathcal{X}_\varepsilon$ is less than a C and therefore $d = 0$.

Even more generally, one can even define the semi-metric ℓ according to the behavior of f around x^* in order that (3) holds. For example if the space $\mathcal{X}$ is a normed space (with norm $\|\cdot\|$), one can define the metric $\ell(x,y) \overset{\text{def}}{=} \tilde{\ell}(\|x-y\|)$ for any $r \ge 0$ as:

$$\tilde{\ell}(r) = \sup_{x;\|x^*-x\|\le r} [f(x^*) - f(x)]\,.$$

Thus $f(x^*) - \ell(x,x^*)$ naturally forms a lower-envelope of f. Thus assuming that the first inequality of (11) (upper-envelope) holds, then $d = 0$ again.

However, although the case $d = 0$ is quite general, it does not hold in situations where there is a discrepancy between the upper- and lower-envelopes (Figure 3).

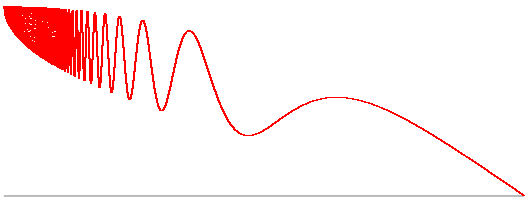

Figure 3. We illustrate the case of a function with different order in the upper and lower envelopes, when $\ell(x,y) = \|x-y\|^\alpha$. Here $f(x) = 1 - \sqrt{x} + (-x^2 + \sqrt{x})\cdot(\sin(1/x^2) + 1)/2$. The lower-envelope behaves like a square root whereas the upper one is quadratic. The maximum number of ℓ-balls with radius ε that can pack $\mathcal{X}_\varepsilon$ (i.e., ℓ_2-balls with radius $\varepsilon^{1/\alpha}$) is at most $\varepsilon^{1/2}/\varepsilon^{1/\alpha} \le \varepsilon^{-3/2}$, since $\alpha \le 1/2$ in order to satisfy (3). We deduce that there is no semi-metric of the form $\|x-y\|^\alpha$ for which $d < 3/2$.

6. Experiments

In this section we numerically evaluate the performance of `StoSOO`[4]. In all experiments with set the parameters k, δ, and $h_{\max}$ to the values from Corollary 2.

[4] code at http://researchers.lille.inria.fr/~valko/hp/project-stosoo

Moreover, we set the branching factor to $K = 3$. Note that when the branching factor is an odd number ($K \geq 3$), we can reuse the evaluations (samples) from the parent node. Indeed, if K is odd, the representative point of the parent node $\circ[h, i]$ will have the same value as the middle child $\circ[h+1, (K+1)/2]$, i.e., $x_{h,i} = x_{h+1,i_{(K+1)/2}}$. In the case when the domain of f is multi-dimensional, we only need to split along one dimension at the time, when expanding the node. In order to preserve bounded diameters assumption, we can split each time along the dimension in which the cell is the largest.

For the evaluation we added a truncated (so that rewards are bounded) zero mean Gaussian noise $\mathcal{N}_T$, sample of which is shown in Figure 4. In all the experiments we performed 10 trials and the error bars in the figures correspond to standard deviations.

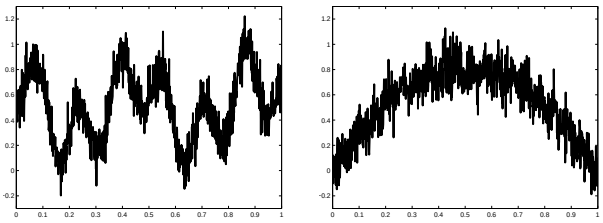

Figure 4. Functions from Figure 1 noised with $\mathcal{N}_T(0, 0.1)$.

Two-sine product: In the first set of experiments we consider a two-sine product function displayed in Figure 1 (left) maximized for $f(0.867526) \approx 0.975599$. Figure 5 displays the performance of `StoSOO` for different levels of noise. We observe that as we increase the number of evaluations, the regret of `StoSOO` decreases.

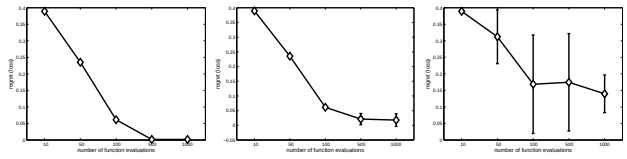

Figure 5. `StoSOO`'s performance for function f_1. **Left**: Noised with $\mathcal{N}_T(0, 0.01)$. **Middle**: Noised with $\mathcal{N}_T(0, 0.1)$. **Right**: Noised with $\mathcal{N}_T(0, 1)$.

In Figure 6, we compare `StoSOO` to the straightforward stochastic version of DOO (Munos, 2011), where we expand each node after $\log(n^2/\delta)/(2w(h)^2)$ evaluations (i.e. when the size of the confidence interval becomes smaller than the diameter $w(h)$ of the cell).

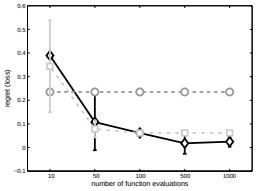

Figure 6. `StoSOO` (diamonds) vs. Stochastic DOO with ℓ_1 (circles) and ℓ_2 (squares) on f_1.

However, (stochastic) DOO needs to know the semi-metric ℓ in order to define $w(h)$. We evaluate the performance of this stochastic DOO using two semi-metrics that satisfy Assumption A1: $\ell_1(x, y) = 12|x - y|$ (for which $d = 1/2$) and $\ell_2(x, y) = 144|x - y|^2$ (for which $d = 0$). We observe that `StoSOO` performs as well as stochastic DOO for the better metric *without* the knowledge of it.

Garland function: Next, we consider a *garland* function displayed in Figure 1 (right). The optimization of this function is challenging because f_2 is not Lipschitz for any L. However its near-optimality dimension is still $d = 0$ (Section 5.2). Figure 7 shows the performance of `StoSOO` as we vary the number of the evaluations. Notice a higher variance at iteration 200 in the left plot; this is because for that many iterations, `StoSOO` was able to reach the depth $h = 6$ but only for a few nodes (while only $h = 5$ for less iterations) with small number of $\lceil 200/(\log^3(200)) \rceil = 2$ evaluations.

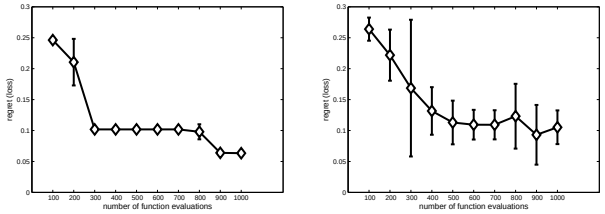

Figure 7. `StoSOO`'s performance for the garland function. **Left** noised with $\mathcal{N}_T(0, 0.01)$. **Right**: Noised with $\mathcal{N}_T(0, 0.1)$.

7. Conclusion

We presented the `StoSOO` algorithm that is able to optimize black-box stochastic functions, without the knowledge of their smoothness. We derived a finite-time performance bound on the expected loss for the important case when there exists a semi-metric such that the near-optimality dimension $d = 0$. We showed that this case corresponds to a large class of functions. In such cases, the performance is almost as good as with an algorithm that would know the best valid semi-metric. In the future we plan to derive finite-time performance for the case $d > 0$.

8. Acknowledgements

This research work presented in this paper was supported by European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 270327 (project CompLACS).

References

Auer, Peter, Cesa-Bianchi, Nicolò, and Fischer, Paul. Finite-time Analysis of the Multiarmed Bandit Problem. *Machine Learning*, 47(2-3):235–256, 2002.

Bubeck, Sébastien, Munos, Rémi, Stoltz, Gilles, and Szepesvári, Csaba. Online Optimization of X-armed Bandits. In *Advances in Neural Information Processing Systems*, pp. 201–208, 2008.

Bubeck, Sébastien, Munos, Rémi, and Stoltz, Gilles. Pure Exploration in Multi-armed Bandits Problems. *Algorithmic Learning Theory*, pp. 23–37, 2009.

Bubeck, Sébastien, Munos, Rémi, Stoltz, Gilles, and Szepesvari, Csaba. X-armed bandits. *Journal of Machine Learning Research*, 12:1587–1627, 2011a.

Bubeck, Sébastien, Stoltz, Gilles, and Yuan, Yu-Jia. Lipschitz bandits without the Lipschitz constant. In *Algorithmic Learning Theory*, pp. 144–158. Springer, 2011b.

Bull, Adam. Convergence rates of efficient global optimization algorithms. *The Journal of Machine Learning Research*, 12:2879–2904, 2011.

Coquelin, Pierre-Arnaud and Munos, Rémi. Bandit Algorithms for Tree Search. In *Uncertainty in Artificial Intelligence*, pp. 67–74, 2007.

Gelly, Sylvain, Kocsis, Levente, Schoenauer, Marc, Sebag, Michèle, Silver, David, Szepesvári, Csaba, and Teytaud, Olivier. The grand challenge of computer Go: Monte Carlo tree search and extensions. *Commun. ACM*, 55(3):106–113, March 2012.

Hansen, Eldon and Walster, William. *Global Optimization Using Interval Analysis: Revised and Expanded*. Pure and Applied Mathematics Series. Marcel Dekker, 2004.

Hren, Jean-Francois and Munos, Rémi. Optimistic Planning of Deterministic Systems. In *European Workshop on Reinforcement Learning*, pp. 151–164, 2008.

Jones, David, Perttunen, Cary, and Stuckman, Bruce. Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Applications*, 79(1):157–181, 1993.

Kearfott, R Baker. *Rigorous Global Search: Continuous Problems*. Nonconvex Optimization and Its Applications. Springer, 1996.

Kleinberg, Robert, Slivkins, Alexander, and Upfal, Eli. Multi-armed bandit problems in metric spaces. In *Proceedings of the 40th ACM symposium on Theory Of Computing*, pp. 681–690, 2008.

Kocsis, Levente and Szepesvári, Csaba. Bandit based Monte-Carlo Planning. In *Proceedings of the 15th European Conference on Machine Learning*, pp. 282–293. Springer, 2006.

Munos, Rémi. Optimistic Optimization of Deterministic Functions without the Knowledge of its Smoothness. In *Advances in Neural Information Processing Systems*, pp. 783–791, 2011.

Neumaier, Arnold. *Interval Methods for Systems of Equations*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2008.

Osborne, Michael. *Bayesian Gaussian processes for sequential prediction, optimisation and quadrature*. PhD thesis, University of Oxford, 2010.

Pintér, János. *Global Optimization in Action: Continuous and Lipschitz Optimization: Algorithms, Implementations and Applications*. Nonconvex Optimization and Its Applications. Springer, 1995.

Slivkins, Aleksandrs. Multi-armed bandits on implicit metric spaces. In *Advances in Neural Information Processing Systems 24*, pp. 1602–1610. 2011.

Srinivas, Niranjan, Krause, Andreas, Kakade, Sham, and Seeger, Matthias. Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design. *Proceedings of International Conference on Machine Learning*, pp. 1015–1022, 2010.

Strongin, Roman and Sergeyev, Yaroslav. *Global Optimization with Non-Convex Constraints: Sequential and Parallel Algorithms*. Nonconvex Optimization and Its Applications. Springer, 2000.