

# MVA Reinforcement Learning

## Optimization of very difficult functions

Nathan de Lara, Florian Tilquin

January 6, 2016

## 1 Introduction

### 1.1 Problem statement

The goal of this paper is to test and compare recently developed algorithms for the optimization of *very difficult functions*. This work is based on the papers by Bubeck [2], Grill [4], Lazaric [1], Bull [3] and Valko [5]. Each one of this paper has a specific definition of *difficult* but the general idea is that the function to optimize has many local maxima and only one global maximum, it has very fast variations and is not necessarily differentiable such that a gradient-based approach to find the optimum should not be successful. All functions are assumed to be bounded and to have a compact support which, up to scaling can be fixed to be  $[0, 1]^p$ . In this paper, we only consider  $p = 1$ . In the end, the general formulation of the problem is:

$$\text{maximize } f(x) \text{ for } x \in [0, 1] \quad (1)$$

### 1.2 About the multi-armed bandit

As previously mentioned, a gradient-based approach is not likely to perform well on the considered functions. Thus, the idea is to use a multi-armed bandit with theoretically an infinite number of arms, sometimes called *continuum-armed bandit*. The algorithms progressively defines a sequence of evaluation points  $x_t$  and observes a reward  $y_t = f(x_t) + \xi_t$  where  $\xi_t$  is a noise term. The choice of  $x_{t+1}$  depends on the sequence of  $(y_{t'})_{t' \leq t}$  and is meant to converge to  $x^*$  while controlling the cumulative regret:

$$R_T = \sum_{t=1}^T f(x^*) - f(x_t) \quad (2)$$

In practice, for the algorithms considered,  $x_t = \mathcal{U}(I_t)$ . This means that  $x_t$  is pulled uniformly at random in an interval  $I_t$  that is chosen by the algorithm. The set of intervals defines a tree structure such that the set of leaves is a partition of  $[0, 1]$ . In the end, the original problem is transformed into a particular case of regular multi-armed bandit. Without any prior information on the function to optimize, we only consider here the family of *dyadic trees*. As mentioned in [3], the dyadic tree on  $[0, 1]$  is the tree with root node  $[0, 1]$ , and where each node  $[a, b]$  has children  $[a, \frac{1}{2}(a+b)]$ ,  $[\frac{1}{2}(a+b), b]$ . An example of dyadic tree is displayed in 1.2.

## 2 Algorithms

In this section, we list the algorithms to be compared and briefly present their respective behaviors. The reader is welcome to consult the original papers for more detailed descriptions and proofs on theoretical performances. As previously mentioned, each algorithm defines a sequence of intervals  $I_t$  from which  $x_t$  is sampled. Typically, the selected interval maximizes a certain selection function of  $(x_{t'}, y_{t'})_{t' < t}$  among a subset of considered intervals. The main differences in the following algorithms is the definition of the selection function and the choice of the considered intervals at each time step.

### 2.1 Hierarchical Optimistic Optimization

### 2.2 Parallel Optimistic Optimization

### 2.3 High-Confidence Tree

### 2.4 Stochastic Simultaneous Optimistic Optimization

### 2.5 Adaptive-Treed Bandits

## 3 Results

### 3.1 Experimental Setup

**Objective functions** We test the algorithms on different reference functions from [5] and [4]:

1. Two-sine product function:  $f_1(x) = \frac{1}{2}(\sin(13x) \cdot \sin(27x)) + 0.5$ .
2. Garland function:  $f_2(x) = 4x(1-x) \cdot (\frac{3}{4} + \frac{1}{4}(1 - \sqrt{|\sin(60x)|}))$ .
3. Grill function:  $f_3(x) = s(\log_2(|x-0.5|)) \cdot (\sqrt{|x-0.5|} - (x-0.5)^2) - \sqrt{|x-0.5|}$   
where  $s(x) = \mathbf{1}(x - \lfloor x \rfloor \in [0, 0.5])$ .

The associated plots are displayed in 3.1.

**Algorithms setup** In order to compare the performances of the different algorithms, we set a desired precision  $\epsilon$  and a total number of function evaluations  $T$  and a number of runs  $N$ . Then we compute for each and each algorithm run the best value returned  $\hat{x}^*$  and the cumulative regret defined in 2. The run is considered a success if  $|\hat{x}^* - x^*| \leq \epsilon$ . The success rates are average cumulative regrets are displayed in 3.1. The tuning of the parameters specific to each algorithm is performed manually.

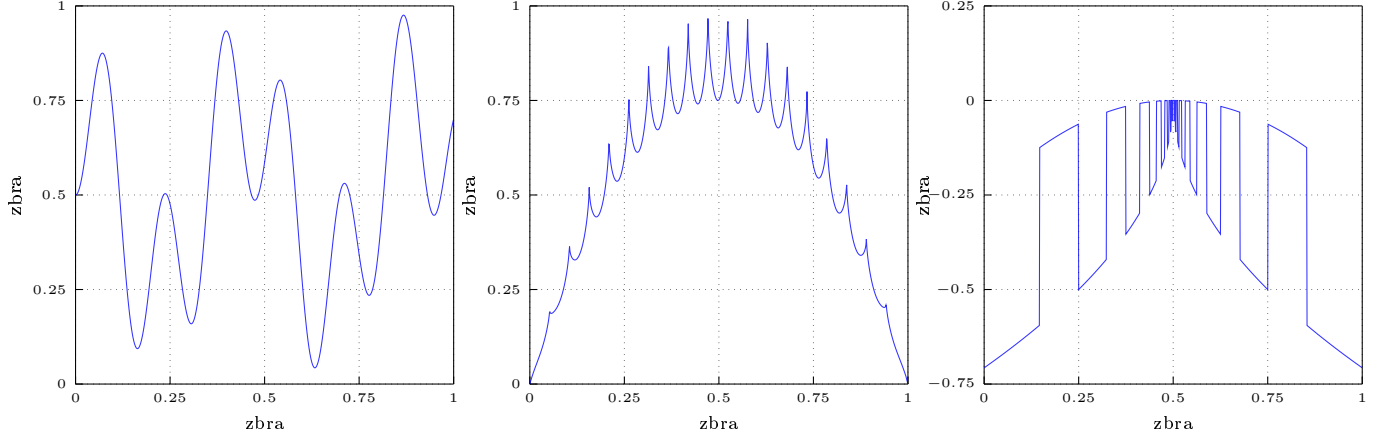


Figure 1: From the left to the right: Two-sine product, Garlang, Grill.

#### Reference functions to optimize

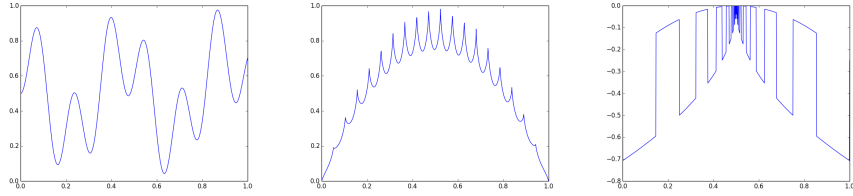


Figure 2: From the left to the right: Two-sine product, Garlang, Grill.

#### Success rates and average cumulative regrets of the algorithms

Algorithm	$f_1$	$f_2$	$f_3$	$R_1$	$R_2$	$R_3$
HOO						
POO						
HCT						
StoSOO						
ATB						

Figure 3: These results are obtained for  $\epsilon =$ ,  $T =$  and  $N =$ .

---

**Algorithm 1** the hierarchical optimistic optimization algorithm
 

---

```

1: for  $n = 1 \dots N$  do
2:    $P \leftarrow [0, 1]$ 
3:    $h, i \leftarrow 0, 1$ 
4:   while  $[h, i] \in \mathcal{T}$  do
5:     if  $B_{h+1, 2i-1} > B_{h+1, 2i}$  then
6:        $h, i \leftarrow h + 1, 2i - 1$ 
7:     else if  $B_{h+1, 2i-1} < B_{h+1, 2i}$  then
8:        $h, i \leftarrow h + 1, 2i$ 
9:     else
10:       $Z \sim \text{Ber}(0.5)$ 
11:       $h, i \leftarrow h + 1, 2i - Z$ 
12:    end if
13:     $P \leftarrow P \cup [h, i]$ 
14:  end while
15:   $H, I \leftarrow h, i$ 
16:   $X \sim \mathcal{U}(\mathcal{P}_{H, I})$ 
17:  reward:  $Y = f(X)$ 
18:   $\mathcal{T} \leftarrow \mathcal{T} \cup [H, I]$ 
19:  for all  $[h, i] \in P$  do
20:     $T_{h, i} \leftarrow T_{h, i} + 1$ 
21:     $\mu_{h, i} \leftarrow (1 - \frac{1}{T_{h, i}})\mu_{h, i} + \frac{Y}{T_{h, i}}$ 
22:  end for
23:  for all  $[h, i] \in \mathcal{T}$  do
24:     $U_{h, i} \leftarrow \mu_{h, i} + \sqrt{\frac{2 \ln n}{T_{h, i}}} + \mu_1 \rho^h$ 
25:  end for
26:   $B_{H+1, 2I-1}, B_{H+1, 2I} \leftarrow +\infty, +\infty$ 
27:   $\mathcal{T}' \leftarrow \mathcal{T}$ 
28:  while  $\mathcal{T}' \neq \{[0, 1]\}$  do
29:     $[h, i] \leftarrow \text{Leaf}(\mathcal{T}')$ 
30:     $B_{h, i} \leftarrow \min\{U_{h, i}, B_{h+1, 2i-1}, B_{h+1, 2i}\}$ 
31:     $\mathcal{T}' \leftarrow \mathcal{T}' \setminus [h, i]$ 
32:  end while
33: end for

```

---

---

**Algorithm 2** the hierarchical optimistic optimization algorithm

---

```
1: for  $n = 1 \dots N_{ev}$  do
2:    $P \leftarrow [0, 1]$ 
3:    $h, i \leftarrow 0, 1$ 
4:   while  $[h, i] \in \mathcal{T}$  do
5:     if  $B_{h+1, 2i-1} > B_{h+1, 2i}$  then
6:        $h, i \leftarrow h + 1, 2i - 1$ 
7:     else if  $B_{h+1, 2i-1} < B_{h+1, 2i}$  then
8:        $h, i \leftarrow h + 1, 2i$ 
9:     else
10:       $Z \sim \text{Ber}(0.5)$ 
11:       $h, i \leftarrow h + 1, 2i - Z$ 
12:    end if
13:     $P \leftarrow P \cup [h, i]$ 
14:  end while
15:   $H, I \leftarrow h, i$ 
16:   $X \sim \mathcal{U}(\mathcal{P}_{H, I})$ 
17:  reward:  $Y = f(X)$ 
18:   $\mathcal{T} \leftarrow \mathcal{T} \cup [H, I]$ 
19:  for all  $[h, i] \in P$  do
20:     $T_{h, i} \leftarrow T_{h, i} + 1$ 
21:     $\mu_{h, i} \leftarrow (1 - \frac{1}{T_{h, i}})\mu_{h, i} + \frac{Y}{T_{h, i}}$ 
22:  end for
23:  for all  $[h, i] \in \mathcal{T}$  do
24:     $U_{h, i} \leftarrow \mu_{h, i} + \sqrt{\frac{2 \ln n}{T_{h, i}}} + \mu_1 \rho^h$ 
25:  end for
26:   $B_{H+1, 2I-1}, B_{H+1, 2I} \leftarrow +\infty, +\infty$ 
27:   $\mathcal{T}' \leftarrow \mathcal{T}$ 
28:  while  $\mathcal{T}' \neq \{[0, 1]\}$  do
29:     $[h, i] \leftarrow \text{Leaf}(\mathcal{T}')$ 
30:     $B_{h, i} \leftarrow \min\{U_{h, i}, B_{h+1, 2i-1}, B_{h+1, 2i}\}$ 
31:     $\mathcal{T}' \leftarrow \mathcal{T}' \setminus [h, i]$ 
32:  end while
33: end for
```

---

### 3.2 Analysis

## A S'more random shit in the appendix

### A.1 HOO

### A.2 POO

### A.3 SOO

### A.4 HCT

### A.5 ATB

## References

- [1] Mohammad Gheshlaghi Azar, Alessandro Lazaric, and Emma Brunskill. On-line stochastic optimization under correlated bandit feedback. *arXiv preprint*

---

**Algorithm 3** the hierarchical optimistic optimization algorithm

---

```

1: for  $n = 1 \dots N_{ev}$  do
2:    $P \leftarrow [0, 1]$ 
3:    $h, i \leftarrow 0, 1$ 
4:   while  $[h, i] \in \mathcal{T}$  do
5:     if  $B_{h+1,2i-1} > B_{h+1,2i}$  then
6:        $h, i \leftarrow h + 1, 2i - 1$ 
7:     else if  $B_{h+1,2i-1} < B_{h+1,2i}$  then
8:        $h, i \leftarrow h + 1, 2i$ 
9:     else
10:       $Z \sim \text{Ber}(0.5)$ 
11:       $h, i \leftarrow h + 1, 2i - Z$ 
12:    end if
13:     $P \leftarrow P \cup [h, i]$ 
14:  end while
15:   $H, I \leftarrow h, i$ 
16:   $X \sim \mathcal{U}(\mathcal{P}_{H,I})$ 
17:  reward:  $Y = f(X)$ 
18:   $\mathcal{T} \leftarrow \mathcal{T} \cup [H, I]$ 
19:  for all  $[h, i] \in P$  do
20:     $T_{h,i} \leftarrow T_{h,i} + 1$ 
21:     $\mu_{h,i} \leftarrow (1 - \frac{1}{T_{h,i}})\mu_{h,i} + \frac{Y}{T_{h,i}}$ 
22:  end for
23:  for all  $[h, i] \in \mathcal{T}$  do
24:     $U_{h,i} \leftarrow \mu_{h,i} + \sqrt{\frac{2 \ln n}{T_{h,i}}} + \mu_1 \rho^h$ 
25:  end for
26:   $B_{H+1,2I-1}, B_{H+1,2I} \leftarrow +\infty, +\infty$ 
27:   $\mathcal{T}' \leftarrow \mathcal{T}$ 
28:  while  $\mathcal{T}' \neq \{[0, 1]\}$  do
29:     $[h, i] \leftarrow \text{Leaf}(\mathcal{T}')$ 
30:     $B_{h,i} \leftarrow \min\{U_{h,i}, B_{h+1,2i-1}, B_{h+1,2i}\}$ 
31:     $\mathcal{T}' \leftarrow \mathcal{T}' \setminus [h, i]$ 
32:  end while
33: end for

```

---

*arXiv:1402.0562*, 2014.

- [2] Sébastien Bubeck, Rémi Munos, Gilles Stoltz, and Csaba Szepesvari. X-armed bandits. *The Journal of Machine Learning Research*, 12:1655–1695, 2011.
- [3] Adam D Bull. Adaptive-treed bandits. *arXiv preprint arXiv:1302.2489*, 2013.
- [4] Jean-Bastien Grill, Michal Valko, and Rémi Munos. Black-box optimization of noisy functions with unknown smoothness. In *Neural Information Processing Systems*, 2015.

---

**Algorithm 4** the hierarchical optimistic optimization algorithm

---

```
1: for  $n = 1 \dots N_{ev}$  do
2:    $P \leftarrow [0, 1]$ 
3:    $h, i \leftarrow 0, 1$ 
4:   while  $[h, i] \in \mathcal{T}$  do
5:     if  $B_{h+1,2i-1} > B_{h+1,2i}$  then
6:        $h, i \leftarrow h + 1, 2i - 1$ 
7:     else if  $B_{h+1,2i-1} < B_{h+1,2i}$  then
8:        $h, i \leftarrow h + 1, 2i$ 
9:     else
10:       $Z \sim \text{Ber}(0.5)$ 
11:       $h, i \leftarrow h + 1, 2i - Z$ 
12:    end if
13:     $P \leftarrow P \cup [h, i]$ 
14:  end while
15:   $H, I \leftarrow h, i$ 
16:   $X \sim \mathcal{U}(\mathcal{P}_{H,I})$ 
17:  reward:  $Y = f(X)$ 
18:   $\mathcal{T} \leftarrow \mathcal{T} \cup [H, I]$ 
19:  for all  $[h, i] \in P$  do
20:     $T_{h,i} \leftarrow T_{h,i} + 1$ 
21:     $\mu_{h,i} \leftarrow (1 - \frac{1}{T_{h,i}})\mu_{h,i} + \frac{Y}{T_{h,i}}$ 
22:  end for
23:  for all  $[h, i] \in \mathcal{T}$  do
24:     $U_{h,i} \leftarrow \mu_{h,i} + \sqrt{\frac{2 \ln n}{T_{h,i}}} + \mu_1 \rho^h$ 
25:  end for
26:   $B_{H+1,2I-1}, B_{H+1,2I} \leftarrow +\infty, +\infty$ 
27:   $\mathcal{T}' \leftarrow \mathcal{T}$ 
28:  while  $\mathcal{T}' \neq \{[0, 1]\}$  do
29:     $[h, i] \leftarrow \text{Leaf}(\mathcal{T}')$ 
30:     $B_{h,i} \leftarrow \min\{U_{h,i}, B_{h+1,2i-1}, B_{h+1,2i}\}$ 
31:     $\mathcal{T}' \leftarrow \mathcal{T}' \setminus [h, i]$ 
32:  end while
33: end for
```

---

- [5] Michal Valko, Alexandra Carpentier, and Rémi Munos. Stochastic simultaneous optimistic optimization. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 19–27, 2013.