OPT2021: 13th Annual Workshop on Optimization for Machine Learning 1–12

Optimum-statistical Collaboration Towards General and Efficient Black-box Optimization

Wenjie Li * LI3549@PURDUE.EDU
Chi-Hua Wang * WANG3667@PURDUE.EDU
Guang Cheng CHENGG@PURDUE.EDU
Department of Statistics, Purdue University

Abstract

We propose a general optimum-statistical collaboration framework for sequential black-box optimization. Based on general definitions of the resolution descriptor and the uncertainty quantifier, we provide a general regret analysis of the proposed framework. We then show that the proposed framework can be applied to a broader range of functions that have different smoothness, and it inspires tighter measures of the statistical uncertainty and thus a faster algorithm.

1. Introduction

We study the problem of sequentially optimizing a black-box objective function $f : \mathcal{X} \mapsto \mathbb{R}$, where $\mathcal{X}$ is the parameter space. The sampling budget (number of evaluations) is denoted by an *unknown* constant n, which is often limited when the cost of evaluating $f(x)$ is expensive. At each round (evaluation) t, the algorithm selects a value $x_t \in \mathcal{X}$ and receives an stochastic feedback $r_t \in [0, 1]$, modelled by $r_t \equiv f(x_t)+\epsilon_t$, where the noise ϵ_t is assumed to be mean zero, bounded by $[-\frac{b}{2}, \frac{b}{2}]$ for some constant $b > 0$, and independent from the historical observed algorithm performance and the path of selected x_t's. We assume that there exists at least one point $x^* \in \mathcal{X}$ such that it attains the maximum f^*, i.e., $f^* \equiv f(x^*) = \sup_{x\in\mathcal{X}} f(x)$. The goal of a black-box optimization algorithm is to gradually find x_n such that $f(x_n)$ is close to the global maximum f^*.

2. Preliminaries

Regret Analysis Framework. Given the measurable space $\mathcal{X}$ and the unknown function f, our objective is to find the maximum of f with a total of n evaluations. We measure the performance of different algorithms using the *cumulative regret.* With respect to the optimal value f^*, the *cumulative regret* of a black-box optimization algorithm is defined as $R_n \equiv nf^* - \sum_{t=1}^{n} r_t$.

Hierarchical partitioning. We use the hierarchical partitioning $\mathcal{P} = \{\{\mathcal{P}_{h,i}\}_{i=1}^{|\mathcal{I}_h|}\}_{h=0}^{\infty}$ to discretize the hyperparameter space $\mathcal{X}$ into cells (nodes), which are previously studied in Munos (2011); Bubeck et al. (2011); Valko et al. (2013). For any non-negative integer h, the set $\{\mathcal{P}_{h,i}\}_{1\le i\le|\mathcal{I}_h|}$ partitions the whole space $\mathcal{X}$, where $\mathcal{I}_h$ is the set of nodes at each depth level h. At depth $h = 0$, a single node $\mathcal{P}_{0,1}$ covers the entire space. Every time we increase the level of depth, each node at the current depth level will be separated into two children; that is, $\mathcal{P}_{h,i} = \mathcal{P}_{h+1,2i-1} \cup \mathcal{P}_{h+1,2i}$. Such a hierarchical partition naturally inspires tree-based algorithms which explores the space by traversing the partitions and selecting the nodes with higher rewards to form a tree structure, with $\mathcal{P}_{0,1}$ being the root.Similar to Grill et al. (2015), we name the partition where each cell is split into regular same-sized subcells as the standard partitioning.

We introduce the generalized definition of near-optimality dimension, which is a natural generalization of the near-optimality notion introduced by Grill et al. (2015).

Near-optimality dimension. For any constants $\alpha > 0, C > 1$, and any function $\xi(h)$ that satisfies $\forall h \geq 1, \xi(h) \in (0,1]$, we define the near-optimality dimension of f with respect to $\mathcal{P}$ as

$$d = d(\alpha, C, \xi(h)) \equiv \inf\{d' > 0 : \forall h \geq 0, \mathcal{N}_h(\alpha\xi(h)) \leq C\xi(h)^{-d'}\}$$

where $\mathcal{N}_h(\epsilon)$ is the number of cells $\mathcal{P}_{h,i}$ on level h such that $\sup_{x\in\mathcal{P}_{h,i}} f(x) \geq f^* - \epsilon$

In simpler words, for each $h > 0$, $\mathcal{N}_h(\alpha\xi(h))$ is the number of near-optimal regions that are $(\alpha\xi(h))$-close to the optimum so that any algorithm has to sample from these regions. $d = d(\alpha, C, \xi(h))$ upper bounds the logarithm of these quantities on the whole partition $\mathcal{P}$ with respect to the function $\xi(h)$. Note that the above general definition of d covers the near optimality dimension defined by Grill et al. (2015) by simply letting $\xi(h) = \rho^h$ and the coefficient $\alpha = 2\nu$ for some constants $\nu > 0$ and $\rho \in (0,1)$.

Notations. At round t, we use $H(t)$ to represent the maximum depth level explored in the partition by an algorithm. For each node $\mathcal{P}_{h,i}$, we use $T_{h,i}(t)$ to denote the number of times it has been pulled and $r^k(x_{h,i})$ to denote the k-th reward observed for the node.

3. Optimum-statistical Collaboration

In this section we first define two quantities that play important roles in our framework. We then introduce the general optimum-statistical collaboration algorithm and provide its theoretical analysis.

Definition 1 ***(Resolution Descriptor)*** *Define* $\texttt{OE}_h$ *to be the resolution for each layer* h*, which is a function that upper-bounds the change of* f *around the optimum and measures the current optimization error, i.e., for any global optimum* x^*

$$\forall h \geq 0, \forall x \in \mathcal{P}_{h,i_h^*}, f(x) \geq f(x^*) - \texttt{OE}_h \tag{OE}$$

where $\mathcal{P}_{h,i_h^*}$ *is the node on layer* h *in the partition that contain the global optimum* x^**.*

Definition 2 ***(Uncertainty Quantifier)*** *Define* $\texttt{SE}_{h,i}(t)$ *to be the uncertainty estimate for each node* $\mathcal{P}_{h,i}$ *at time* t*, which is a function that upper-bounds the current statistical error of the average* $\widehat{\mu}_{h,i}(t)$ *of the rewards obtained with high probability, i.e., the event*

$$\mathcal{A}_t = \left\{\forall (h,i), |\widehat{\mu}_{h,i}(t) - f(x_{h,i})| \leq \texttt{SE}_{h,i}(t)\right\} \tag{SE}$$

is a high probability event.

Given the above definitions of the resolution descriptor and the uncertainty quantifier at each node, we introduce the optimum-statistical collaboration algorithm in Algorithm 1

Algorithm 1 **Optimum-statistical Collaboration**

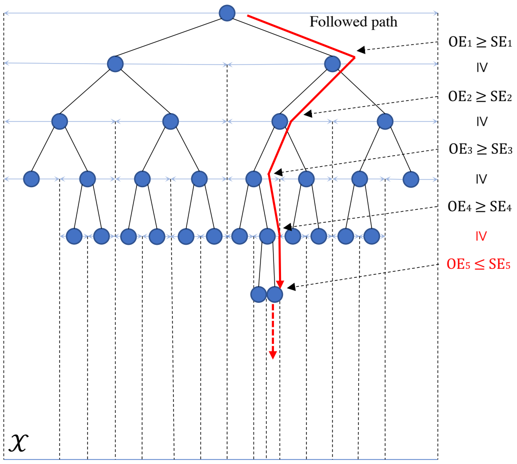
```
Input: Hierachical partition $\mathcal{P}$, resolution descriptor $\mathtt{OE}_h$, Uncertainty quantifier $\mathtt{SE}_{h,i}(t)$.

Step 1: Refresh the confidence at some specific time steps to update all $\mathtt{SE}_{h,i}(t)$ in the tree.
Step 2: Find the optimal node $\mathcal{P}_{h_t,i_t}$ at time $t$ that satisfies $\mathtt{OE}_{h_t} \leq \mathtt{SE}_{h_t,i_t}(t)$ and pull $\mathcal{P}_{h_t,i_t}$.
Step 3: If $\mathtt{OE}_{h_t} \geq \mathtt{SE}_{h_t,i_t}(t)$ after the pull, expand $\mathcal{P}_{h_t,i_t}$ and explore deeper.
```

that guides the design of different tree-based algorithms, with possibly different choices of $\mathtt{OE}$ and $\mathtt{SE}$.

The first step of the collaboration is to update the confidence in the algorithm at the refresh points so that $\mathtt{SE}_{h,i}(t)$ can better measure the statistical uncertainty at each node. The second step and the third step play the most important roles in different algorithms to manage the flux of $\mathtt{SE}_{h,i}(t)$ and $\mathtt{OE}_{h,i}(t)$. The end-goal of the optimum-statistical collaboration is that, after pulling enough number of times, the following relationship holds along the shortest path from the root to the deepest node that contains the global maximum. If there are multiple global maximizers, the process only needs to find one of them.

$$\mathtt{OE}_1 \geq \mathtt{SE}_1 > \mathtt{OE}_2 \geq \mathtt{SE}_2 \geq \cdots \geq \mathtt{OE}_h \geq \mathtt{SE}_h \geq \cdots \tag{1}$$

where we have slightly abused the notation of $\mathtt{OE}_h$ and $\mathtt{SE}_h$ to represent the resolution descriptor and the uncertainty quantifier of the h-th node on the selected path, since the node is on layer h in the tree and its index relies on the objective and the algorithm. In other words, the two terms collaborate on the optimization process so that $\mathtt{SE}$ is controlled by $\mathtt{OE}$ in each node of the selected path, and they both become smaller when the exploration algorithm goes deeper. Figure 1 illustrates the above process more clearly using an example tree on the standard partition.


Followed path
$\mathtt{OE}_1 \geq \mathtt{SE}_1$
≥
$\mathtt{OE}_2 \geq \mathtt{SE}_2$
≥
$\mathtt{OE}_3 \geq \mathtt{SE}_3$
≥
$\mathtt{OE}_4 \geq \mathtt{SE}_4$
≥
$\mathtt{OE}_5 \leq \mathtt{SE}_5$
$\mathcal{X}$

We state the following theorem, which is a general regret upper bound with respect to any choice of $\mathtt{SE}_{h,i}(t)$ and $\mathtt{OE}_h$, and any design of policy that follows the optimum statistical collaboration framework, with only a mild condition on the result of the choice in each round.

Figure 1: Illustration of the optimum-statistical collaboration framework. The node on the fifth layer in the path will be pulled because its $\mathtt{OE} \leq \mathtt{SE}$

Theorem 3 (General Regret Bound) *Suppose that under a sequence of probability events* $\{\mathcal{E}_t\}_{t=1,2,\cdots}$, *at each time* t, *the designed policy to select the optimal node* $\mathcal{P}_{h_t,i_t}$ *in Algorithm 1 satisfies* $f^* - f(x_{h_t,i_t}) \leq a \cdot \max\{\mathtt{SE}_{h_t,i_t}(t),\ \mathtt{OE}_{h_t}\}$, *where* $a > 0$ *is an absolute* *constant. Then for any $\overline{H} \in [1, H(n))$ we have the following bound on the expected regret*

$$\mathbb{E}[R_n] \leq \sqrt{2n\log(4n^3)} + \frac{1}{4n^2} + \sum_{t=1}^{n} \mathbb{P}(\mathcal{E}_t^c) + 2aC\sum_{h=1}^{\overline{H}} (\mathtt{OE}_{h-1})^{-\bar{d}} \int_1^{T_h(n)} \max_i \mathtt{SE}_{h,i}(s)ds$$
$$+ a\sum_{\overline{H}+1}^{H(n)} \sum_{i\in\mathcal{I}_h(n)} \int_1^{T_{h,i}(n)} \mathtt{SE}_{h,i}(s)ds$$

where $\bar{d} := d(a, C, \mathtt{OE}_{h-1})$ is the near-optimality dimension defined with respect to a, C, and $\mathtt{OE}_{h-1}$, and $T_h(n) = \max_i T_{h,i}(n)$

4. Specific Examples

Provided the optimum-statistical collaboration framework and its analysis, we discuss the exact formulas of the resolution descriptor and the uncertainty quantifier and elaborate the roles these definitions play in the optimization process.

4.1. The resolution descriptor (Definition 1)

Local Smoothness. Grill et al. (2015) assumed that there exist two constants $\nu_1 > 0, \rho \in (0, 1)$ s.t.

$$\forall h \geq 0, \forall x \in \mathcal{P}_{h,i_h^*}, f(x) \geq f^* - \nu_1\rho^h \tag{2}$$

The above equation states that the function f is $\nu_1\rho^h$-smooth around the maximum at each level h. It has been considered in many prior works such as Shang et al. (2019); Bartlett et al. (2019). The resolution descriptor is naturally taken to be $\mathtt{OE}_h = \nu_1\rho^h$.

However, such a choice of local smoothness, despite really useful for many blackbox objectives, is too restrictive as it serves as a strong requirement on both the objective f and the partition $\mathcal{P}$. Many functions and partitions do not satisfy Eqn. (2). An simple example is the function $g(x) = 1 + 1/(\ln x)$ defined on the domain $(0, 1/e]$ and with the standard partition. It can be easily proved that it is impossible for $g(x)$ and the standard partition to satisfy Eqn. (2) for any given constants $\nu_0 > 0, \rho_0 \in (0, 1)$ because the function decreases too fast around 0. It might be possible to define a particular partition for $g(x)$ that satisfies Eqn. (2). However, such a partition is defined in hindsight since we do not have any knowledge of the function before the optimization.

Notice that $g(x)$ is actually monotone and thus very easy to optimize, but previous analyses that rely on Eqn. (2) cannot be applied to such functions. We therefore introduce our general $\phi(h)$-local smoothness of the objective to analyze functions that have different levels of smoothness.

General Local Smoothness. Assume that there exists a function $\phi(h) \in (0, 1)$ s.t.

$$\forall h \geq 0, \forall x \in \mathcal{P}_{h,i_h^*}, f(x) \geq f(x^*) - \phi(h) \tag{GLS}$$

In the same example $g(x) = 1+1/(\ln x)$, it can be shown that $g(x)$ satisfies Condition (GLS) with $\phi(h) = 2/h$. Therefore, if we take $\mathtt{OE}_h = 2/h$ in our framework and choose $\mathtt{SE}_{h,i}(t)$ as in the next subsection, a valid regret bound can be obtained for $g(x)$, since $d(2, C, 1/h) < \infty$

in this case. Similarly, we can analyze functions and partitions that satisfy Condition (GLS) with different $\phi(h)$ such as $\phi(h) = 1/h^p$, for some $p > 0$ or $\phi(h) = 1/\log h$, as long as the corresponding near-optimality dimension $d(a, C, \phi(h))$ is finite. Given such a generalized definition and the general bound in Theorem 3, we can provide the convergence results for a much larger class of black-box objectives and partitions, including but not limited to those that satisfy Eqn. (2).

4.2. The uncertainty quantifier (Definition 2)

Tracking Statistics. We first define the following tracking statistics which are useful in different choices of $\texttt{SE}$. The *mean estimate* $\widehat{\mu}_{h,i}(t)$ and the *variance estimate* $\widehat{\mathbb{V}}_{h,i}(t)$ at round t are

$$\widehat{\mu}_{h,i}(t) \equiv \frac{1}{T_{h,i}(t)} \sum_{k=1}^{T_{h,i}(t)} r^k(x_{h,i}),\ \widehat{\mathbb{V}}_{h,i}(t) \equiv \frac{1}{T_{h,i}(t)} \sum_{k=1}^{T_{h,i}(t)} \Big(r^k(x_{h,i}) - \widehat{\mu}_{h,i}(t)\Big)^2$$

The variance estimate is defined to be negative infinity when $T_{h,i}(t) \leq 1$ since variance is undefined in such cases. We now discuss two specific choices of $\texttt{SE}$.

Nonadaptive Quantifier ($\texttt{HCT}$) Azar et al. (2014) proposed in their High Confidence Tree ($\texttt{HCT}$) algorithm to use an uncertainty quantifier of the following form

$$\texttt{SE}_{h,i}(t) \equiv bc\sqrt{\frac{\log(1/\tilde{\delta}(t^+))}{T_{h,i}(t)}}$$

where t^+ is the time step to update the confidence, which is a variable related to t (see **step 1** in §A.2) and c is a tuning constant. By Hoeffding's inequality, the above $\texttt{SE}$ is a high-probability upper bound for the uncertainty (Event $\mathcal{A}_t$ in Section 3).

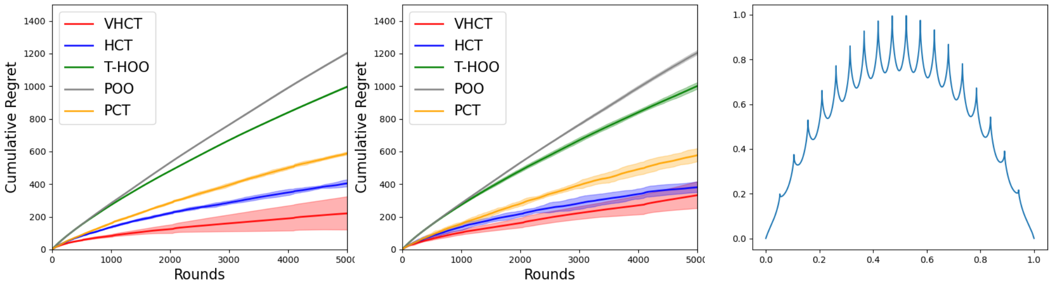
Variance Adaptive Quantifier ($\texttt{VHCT}$) Based on our framework of the statistical collaboration, a tighter measure of the statistical uncertainty can boost the performance of the optimization algorithm, as the goal in Eqn. (1) can be reached faster. We therefore propose the following variance adaptive uncertainty quantifier, which is an adaptive variant of $\texttt{HCT}$.

$$\texttt{SE}_{h,i}(t) \equiv c\sqrt{\frac{2\widehat{\mathbb{V}}_{h,i}(t)\log(1/\tilde{\delta}(t^+))}{T_{h,i}(t)}} + \frac{3bc^2\log(1/\tilde{\delta}(t^+))}{T_{h,i}(t)} \tag{3}$$

The uniqueness of the above $\texttt{SE}_{h,i}(t)$ is that it utilizes the variance of each node. Therefore, when the variance of the stochastic reward is small, the objective $\texttt{SE}_{h,i}(t) \leq \texttt{OE}_h$ is achieved faster at the node $\mathcal{P}_{h,i}$. In such a way, the algorithm is able to adapt to different noises. None of the prior works have utilized the variance information in each node when delineating the node uncertainty, and thus the above choice has an advantage over non-adaptive algorithms. The specific $\texttt{VHCT}$ algorithm is provided in Appendix A

4.3. Regret Bounds

We now provide regret bounds of the algorithm $\texttt{VHCT}$, which serve as examples of our general Theorem 3 when $\texttt{OE}$ and $\texttt{SE}$ are specified. The regret bounds depend on the maximum variance in history across all the nodes $V_{\max} = \max_{h,i,t} \widehat{\mathbb{V}}_{h,i}(t)$. We focus on two choices



(a) Low noise (b) Moderate noise (c) Garland function $G(x)$

Figure 2: Cumulative regret of different algorithms on the Garland function $G(x)$

of the local smoothness function in Condition (GLS) and their corresponding near-optimal dimensions, i.e., $\phi(h) = \nu\rho^h$ that matches previous analyses such as Grill et al. (2015); Shang et al. (2019), and $\phi(h) = 1/h$. For other choices of $\phi(h)$, similar regret upper bounds can also be derived by following Theorem 3.

Theorem 4 *Assume that the objective function f satisfies Condition (GLS) with $\phi(h) = \nu\rho^h$ for two constants $\nu > 0, \rho \in (0, 1)$. The expected cumulative regret of Algorithm 2 is upper bounded by*

$$\mathbb{E}[R_n^{\texttt{VHCT}}] \leq 2\sqrt{2n\log(4n^3)} + C_1 V_{\max}^{\frac{1}{d_1+2}} n^{\frac{d_1+1}{d_1+2}} (\log n)^{\frac{1}{d_1+2}} + C_2 n^{\frac{2d_1+1}{2d_1+4}} \log n.$$

where C_1 and C_2 are two constants and the near-optimality dimension $d_1 = d(3\nu, C, \rho^h)$.

Theorem 5 *Assume that the objective function f satisfies Condition (GLS) with $\phi(h) = 1/h$. The expected cumulative regret of Algorithm 2 is upper bounded by*

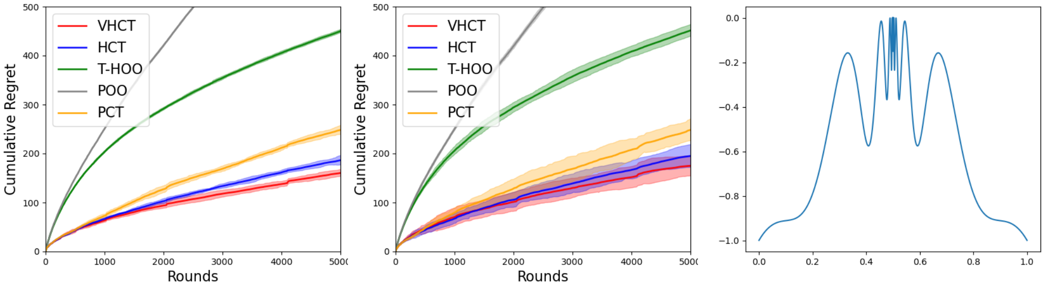
$$\mathbb{E}[R_n^{\texttt{VHCT}}] \leq 2\sqrt{2n\log(4n^3)} + \bar{C}_1 V_{\max}^{\frac{1}{2d_2+3}} n^{\frac{2d_2+2}{2d_2+3}} (\log n)^{\frac{1}{2d_2+3}} + \bar{C}_2 n^{\frac{2d_2+1}{2d_2+3}} \log n.$$

where $\bar{C}_1$ and $\bar{C}_2$ are two constants and the near-optimality dimension $d_2 = d(1, C, 1/h)$.

Remark 6 *Note that these regret bounds depend on the maximum variance appeared in the history, therefore we would expect* `VHCT` *to be faster than the* `HCT` *algorithms when the noise is small.*

5. Experiments

We empirically compare the proposed `VHCT` algorithm with the existing **anytime** blackbox optimization algorithms, including `T-HOO` (the truncated version of `HOO`), `HCT`, `POO`, and `PCT` (`POO` + `HCT`, (Shang et al., 2019)). We use the Garland function and the Double-sine function as the blackbox objectives. We run the algorithms for two settings, i.e., the low noise setting where $\epsilon_t \sim \text{Uniform}(-0.05, 0.05)$, and the moderate noise setting where $\epsilon_t \sim \text{Uniform}(-0.2, 0.2)$. For the algorithms that need the smoothness parameters, we follow



(a) Low noise (b) Moderate noise (c) Double-sine function $D(x)$

Figure 3: Cumulative regret of different algorithms on the Double-sine function $D(x)$

Shang et al. (2019) and tune ρ from $\{0.25, 0.5, 0.75\}$. We report the best result for each algorithm. For POO and PCT, we follow Grill et al. (2015) and use $\rho_{\max} = 0.9$. The predefined bound for the noise is set to be $b = 1$ for all the algorithms. We run every algorithm for 20 independent trials and plot the average cumulative regret with error bounds.

As shown in Figure 2 and 3, VHCT converges much faster than any other algorithms in both the low-noise setting and the moderate-noise setting. Note that the local maximums of $D(x)$ are much closer to its maximum than $G(x)$, therefore the advantage of VHCT is less conspicuous on $D(x)$, but it still remains the fastest algorithm. All the experiment results validate our claims in Section 4.

References

Mohammad Gheshlaghi Azar, Alessandro Lazaric, and Emma Brunskill. Online stochastic optimization under correlated bandit feedback. In *International Conference on Machine Learning*, pages 1557–1565. PMLR, 2014.

Peter L. Bartlett, Victor Gabillon, and Michal Valko. A simple parameter-free and adaptive approach to optimization under a minimal local smoothness assumption. In *30th International Conference on Algorithmic Learning Theory*, 2019.

Sébastien Bubeck, Rémi Munos, Gilles Stoltz, and Csaba Szepesvári. ¡i¿x¡/i¿-armed bandits. *Journal of Machine Learning Research*, 12(46):1655–1695, 2011.

Jean-Bastien Grill, Michal Valko, Remi Munos, and Remi Munos. Black-box optimization of noisy functions with unknown smoothness. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2015.

Rémi Munos. Optimistic optimization of a deterministic function without the knowledge of its smoothness. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011.

Xuedong Shang, Emilie Kaufmann, and Michal Valko. General parallel optimization a without metric. In *Algorithmic Learning Theory*, pages 762–788, 2019.

Michal Valko, Alexandra Carpentier, and Rémi Munos. Stochastic simultaneous optimistic optimization. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 19–27. PMLR, 17–19 Jun 2013.

Appendix A. Proof of the General Bound and the VHCT Algorithm

A.1. Proof of the General Regret Bound in Theorem 3

Proof. We decompose the cumulative regret into two terms that depend on the high probability event $\mathcal{E}$. Denote the simple regret at each iteration t to be $\Delta_t = f^* - r_t$, then we can perform the following decomposition

$$\begin{aligned} R_n = \sum_{t=1}^{n} \Delta_t &= \left(\sum_{t=1}^{n} \Delta_t \mathbb{I}_{\mathcal{E}_t} \right) + \left(\sum_{t=1}^{n} \Delta_t \mathbb{I}_{\mathcal{E}_t^c} \right) = R_n^{\mathcal{E}} + R_n^{\mathcal{E}^c} \\ &= R_n^{\mathcal{E}} + \sum_{t=1}^{n} \mathbb{I}_{\mathcal{E}_t^c} \end{aligned}$$

where the last inequality is because $|\Delta_t| \leq 1$. Now note that the instantaneous regret Δ_t can be written as

$$\Delta_t = f^* - r_t = f^* - f\left(x_{h_t,i_t}\right) + f\left(x_{h_t,i_t}\right) - r_t = \Delta_{h_t,i_t} + \widehat{\Delta}_t$$

which means that the regret under the event $\mathcal{E}_t$ can be decomposed into

$$R_n^{\mathcal{E}} = \sum_{t=1}^{n} \Delta_{h_t,i_t} \mathbb{I}_{\mathcal{E}_t} + \sum_{t=1}^{n} \widehat{\Delta}_t \mathbb{I}_{\mathcal{E}_t} \leq \sum_{t=1}^{n} \Delta_{h_t,i_t} \mathbb{I}_{\mathcal{E}_t} + \sum_{t=1}^{n} \widehat{\Delta}_t = \widetilde{R}_n^{\mathcal{E}} + \widehat{R}_n^{\mathcal{E}}$$

Note that by the definition of $\widehat{\Delta}_t$, it is a bounded martingale difference sequence since $\mathbb{E}[\widehat{\Delta_t} \mid \mathcal{F}_{t-1}] = 0$ and $|\Delta_t| \leq 1$, where $\mathcal{F}_t$ is defined to be the filtration generated up to time t. Therefore by Azuma's inequality on this sequence, we get

$$\widehat{R}_n^{\mathcal{E}} \leq \sqrt{2n \log(\frac{4n^2}{\delta})} \tag{4}$$

with probability $1 - \delta/(4n^2)$. Now the only term left is $\widetilde{R}_n^{\mathcal{E}}$, we bound it as follows.

$$\begin{aligned} \widetilde{R}_n^{\mathcal{E}} = \left(\sum_{t=1}^{n} \Delta_{h_t,i_t} \mathbb{I}_{\mathcal{E}_t} \right) &\leq \left(\sum_{h=1}^{H(n)} \sum_{i \in \mathcal{I}_h(n)} \sum_{t=1}^{n} \Delta_{h,i} \mathbb{I}_{(h_t,i_t)=(h,i)} \mathbb{I}_{\mathcal{E}_t} \right) \\ &\leq \sum_{h=1}^{\overline{H}} \sum_{i \in \mathcal{I}_h(n)} \sum_{t=1}^{n} a\mathtt{SE}_{h,i}(t) + \sum_{\overline{H}+1}^{H(n)} \sum_{i \in \mathcal{I}_h(n)} \sum_{t=1}^{n} a\mathtt{SE}_{h,i}(t) \\ &\leq \underbrace{a \sum_{h=1}^{\overline{H}} \sum_{i \in \mathcal{I}_h(n)} \int_{1}^{T_{h,i}(n)} \mathtt{SE}_{h,i}(s) ds}_{\text{(I)}} + \underbrace{a \sum_{\overline{H}+1}^{H(n)} \sum_{i \in \mathcal{I}_h(n)} \int_{1}^{T_{h,i}(n)} \mathtt{SE}_{h,i}(s) ds}_{\text{(II)}} \end{aligned}$$

where $\bar{H}$ is a constant to be tuned later. The second inequality is because under event $\mathcal{E}_t$ when we select $\mathcal{P}_{h_t,i_t}$, we have $\texttt{SE}_{h_t,i_t}(t) \geq \texttt{OE}_{h_t}$. The first term (I) can be bounded as

$$\begin{aligned}
\text{(I)} &\leq a\sum_{h=1}^{\overline{H}} \sum_{i\in\mathcal{I}_h(n)} \int_1^{T_h(n)} \max_i \texttt{SE}_{h,i}(s)ds \leq a\sum_{h=1}^{\overline{H}} |\mathcal{I}_h(n)| \int_1^{T_h(n)} \max_i \texttt{SE}_{h,i}(s)ds \\
&\leq a\sum_{h=1}^{\overline{H}} 2\mathcal{N}_{h-1}\left(a\texttt{OE}_{h-1}\right) \int_1^{T_h(n)} \max_i \texttt{SE}_{h,i}(s)ds \\
&\leq 2aC\sum_{h=1}^{\overline{H}} \left(\texttt{OE}_{h-1}\right)^{-\bar{d}} \int_1^{T_h(n)} \max_i \texttt{SE}_{h,i}(s)ds
\end{aligned}$$

where $T_h(n) = \max_{i\in\mathcal{I}_h(n)} T_{h,i}(n)$ and $\bar{d} := d(a, C, \texttt{OE}_{h-1})$ is the near-optimality dimension with respect to $(a, C, \texttt{OE}_{h-1})$. The third inequality is because we only expand a node into two children, so $|\mathcal{I}_h(n)| \leq 2|\mathcal{I}^+_{h-1}(n)|$ (Note that we do not have any requirements on the number of children of each node, so the binary tree here can be easily replaced by a K-nary tree with $K \geq 2$). Also since we only select a node (h, i) when its parent (h^p, i^p) satisfies $\texttt{OE} \geq \texttt{SE}$, we have $\mathcal{P}_{h^p,i^p}$ satisfies $f^* - f(x_{h^p,i^p}) \leq a\texttt{OE}_{h^p}$ under $\mathcal{E}$. By the definition of $\mathcal{N}_h(\epsilon)$ in the near-optimality dimension, we have

$$|\mathcal{I}_h(n)| \leq 2|\mathcal{I}^+_{h-1}(n)| \leq 2\mathcal{N}_{h-1}\left(a\texttt{OE}_{h-1}\right)$$

and thus the final upper bound for (I). Therefore for any $\bar{H} \in [1, H(n)]$, we have the expectation of the cumulative regret to be upper bounded by

$$\begin{aligned}
\mathbb{E}[R_n] &= \sum_{t=1}^n \mathbb{E}[\Delta_t] = \mathbb{E}[R_n^{\mathcal{E}}] + \sum_{t=1}^n \mathbb{P}(\mathcal{E}_t^c) \\
&\leq \sqrt{2n\log(4n^3)} + \frac{1}{4n^2}\max(\widehat{R}_n^{\mathcal{E}}) + \sum_{t=1}^n \mathbb{P}(\mathcal{E}_t^c) + \widetilde{R}_n^{\mathcal{E}} \\
&\leq \sqrt{2n\log(4n^3)} + \frac{1}{4n^2} + 2aC\sum_{h=1}^{\overline{H}} \left(\texttt{OE}_{h-1}\right)^{-\bar{d}} \int_1^{T_h(n)} \max_i \texttt{SE}_{h,i}(s)ds \\
&\quad + a\sum_{\overline{H}+1}^{H(n)} \sum_{i\in\mathcal{I}_h(n)} \int_1^{T_{h,i}(n)} \texttt{SE}_{h,i}(s)ds + \sum_{t=1}^n \mathbb{P}(\mathcal{E}_t^c)
\end{aligned}$$

where the second inequality is by taking $\delta = 1/n$. □

A.2. Algorithm Example - VHCT

Based on the proposed framework and its analysis, we propose a new algorithm VHCT as an example of Algorithm 1 and reveal its capability to adapt to different noises. Algorithm 2 provides the pseudo-code of VHCT, with two subroutines Algorithm 3 and 4.

Similar to HCT, the proposed VHCT also maintains an upper-bound $U_{h,i}$ for each node to decide collaborative optimism. In particular, for any node $\mathcal{P}_{h,i}$, the upper-bound $U_{h,i}$ is

Algorithm 2 `VHCT` Algorithm

```
Input: functions $\texttt{OE}_h, \texttt{SE}_{h,i}(t)$, partition $\mathcal{P}$.
Initialize: $\mathcal{T}_t = \{\mathcal{P}_{0,1}, \mathcal{P}_{1,1}, \mathcal{P}_{1,2}\}, U_{1,1}(t) = U_{1,2}(t) = +\infty$
for $t \leftarrow 1$ to $n$ do
    if $t = t^+$ then
        for all nodes $\mathcal{P}_{h,i} \in \mathcal{T}_t$ do
            Compute $U_{h,i}(t)$ from Eqn. (5)                        ▷ Step 1.
        end for
        UpdateBackward($\mathcal{T}_t, t$)
    end if
    $(h_t, i_t) \leftarrow$ PullUpdate($\mathcal{T}_t, t$)                ▷ Step 2.
    if $T_{h_t,i_t}(t) \geq \tau_{h_t,i_t}(t)$ and $\mathcal{P}_{h_t,i_t}$ is a leaf then
        $\mathcal{T}_t \leftarrow \mathcal{T}_t \cup \{\mathcal{P}_{h_t+1,2i_t-1}, \mathcal{P}_{h_t+1,2i_t}\}$       ▷ Step 3.
        $U_{h+1,2i}(t) = U_{h+1,2i-1}(t) = +\infty$
    end if
end for
```

computed directly from the observed reward for pulling $x_{h,i}$ as

$$U_{h,i}(t) \equiv \widehat{\mu}_{h,i}(t) + \texttt{OE}_h + \texttt{SE}_{h,i}(t) \tag{5}$$

with $\texttt{SE}_{h,i}(t)$ defined as in Eqn. (3) and $\texttt{OE}_h$ tuned by the algorithm. We also define the $B_{h,i}(t)$ values to be tighter upper bounds to better utilize the tree structure in the algorithm. Since the maximum upper bound of one node cannot be greater than the maximum of its children, $B_{h,i}(t)$ is defined to be

$$B_{h,i}(t) = \min\left\{U_{h,i}(t), \max\{B_{h+1,2i}(t), B_{h+1,2i-1}(t)\}\right\} \tag{6}$$

if the node has children and $B_{h,i}(t) = U_{h,i}(t)$ if not. Every time $U_{h,i}(t)$ is changed, we update all the $B_{h',i'}(t)$ of the nodes $\mathcal{P}_{h',i'}$ superior to $\mathcal{P}_{h,i}$ in the path backward from the leaf to the root.

We now elaborate the `VHCT` algorithm as a three-step process corresponding to the three steps in the optimum-statistical collaboration framework.

Step 1. Refresh the confidence constant for each node. At the refresh points $t^+ = 2^{\lfloor \log t \rfloor + 1}$, we update the confidence $\tilde{\delta}_t = \min\{1, c_1\delta/t\}$ for each node, where c_1 is a constant. Therefore, the uncertainty quantifiers `SE` for all nodes are refreshed. Then we use the algorithm `UpdateBackward` to compute the B-values from the leaves to the root.

Step 2. Find the current optimal path and pull the optimal node. The best node to pull at the moment is found by the helper function `PullUpdate`, which is the node $\mathcal{P}_{h_t,i_t}$ that has the highest upper bound $B_{h_t,i_t}(t)$ in layer h_t and satisfies $\texttt{OE}_{h_t} \leq \texttt{SE}_{h_t,i_t}(t)$. That means the node is the best node in terms of the collaborative optimism between `OE` and `SE`, but its uncertainty is too large so that the tree cannot be expanded or further explored. After the best node is pulled, all statistics concerned with the node and its parents are updated according to the formulas, using the helper function `UpdateBackward`.

Step 3. Decide whether to expand the optimal node. After pulling the best node, the algorithm decides whether it needs to expand the node and explore further. Recall the

Algorithm 3 PullUpdate

```
Input: a tree $\mathcal{T}_t$, round $t$
Initialize: $(h_t, i_t) \leftarrow (0, 1); S_t \leftarrow \mathcal{P}_{0,1}; T_{0,1}(t) = \tau_0(t) = 1$ ;
while $\mathcal{P}_{h_t,i_t}$ is not a leaf, $T_{h_t,i_t}(t) \geq \tau_{h_t,i_t}(t)$ do
    $j \leftarrow \text{argmax}_{j=0,1}\{B_{h_t+1,2i_t-j}(t)\}$
    $(h_t, i_t) \leftarrow (h_t + 1, 2i_t - j)$
    $S_t \leftarrow S_t \cup \{\mathcal{P}_{h_t,i_t}\}$                    ▷Find the optimal node
end while
Pull $x_{h_t,i_t}$ and get reward $r_t$
$T_{h_t,i_t}(t) \leftarrow T_{h_t,i_t}(t) + 1$
Update $\widehat{\mu}_{h_t,i_t}(t)$, $\widehat{\mathbb{V}}_{h_t,i_t}(t)$
$U_{h_t,i_t}(t) \leftarrow \widehat{\mu}_{h_t,i_t}(t) + \texttt{OE}_{h_t} + \texttt{SE}_{h_t,i_t}(t)$
UpdateBackward$(S_t, t)$
return $(h_t, i_t)$
```

Algorithm 4 UpdateBackward

```
Input: a tree $\mathcal{T}$, round $t$
for $\mathcal{P}_{h,i} \in \mathcal{T}$ backward from each leaf of $\mathcal{T}$ do
    if $\mathcal{P}_{h,i}$ is a leaf of $\mathcal{T}$ then
        $B_{h,i}(t) \leftarrow U_{h,i}(t)$
    else
        Compute $B_{h,i}(t)$ from Eqn. (6)
    end if
    Update the threshold $\tau_{h,i}(t)$
end for
```

interaction between the resolution descriptor $\texttt{OE}_h$ and the uncertainty quantifier $\texttt{SE}_{h,i}(t)$ at depth h_t, we know that the optimal node (if it is a leaf) should only be expanded when $\texttt{SE}_{h,i}(t) \leq \texttt{OE}_h$. As a consequence, we define the variable $\tau_{h,i}$ for each node $\mathcal{P}_{h,i}$ as follows.

$$\tau_{h,i}(t) = \inf_{t \in \mathbb{N}}\{\texttt{SE}_{h,i}(t) \leq \texttt{OE}_h\}$$

Note that given the $\texttt{SE}_{h,i}(t)$ of VHCT in Eqn. (3), the above requirement for $\tau_{h,i}(t)$ is a quadratic equation for any $\texttt{OE}_h$ and thus can be easily solved. For example, when $\texttt{OE}_h = 1/h$, we have

$$\tau_{h,i}(t) = \left\lceil \left(ch + \sqrt{c^2h^2 + \frac{6bc^2h}{\widehat{\mathbb{V}}_{h,i}(t)}} \right)^2 \widehat{\mathbb{V}}_{h,i}(t) \log(1/\tilde{\delta}(t^+)) \right\rceil$$

We would only expand the node $\mathcal{P}_{h_t,i_t}$ into its children when the number of pulls is large enough so that $T_{h_t,i_t} \geq \tau_{h_t,i_t}$. Note that $\tau_{h,i}(t)$ is smaller when the variance is smaller, because the relationship $\texttt{SE}_{h,i}(t) \leq \texttt{OE}_h$ is achieved faster. Therefore, VHCT adapts to the variance through its node-specific threshold and thus explores the parameter space faster than the non-adaptive algorithms. However, we emphasize that VHCT is only an example that satisfies our general framework, and many other algorithms that follow the framework can be proposed.