Procrastinated Tree Search

Black-box Optimization with Delayed, Noisy, and Multi-fidelity Feedback

Junxiong Wang,[1] **Debabrota Basu,**[2] **Immanuel Trummer**[1]

[1] Dept. of Computer Science, Cornell University, USA

[2] Scool Team, Inria, CRIStAL, CNRS, Univ. Lille, France

What's Next?

1. Black-box Optimization: Hierarchical Tree Search
2. Procrastinated Tree Search (PCTS)
   2.1 Adapting to Delayed Feedback
   2.2 Adapting to Noisy Feedback
   2.3 Adapting to Multi-fidelity Feedback
3. Performance Evaluation: Global & Hyperparameter Optimization
4. The Curtain Call: What's Here and What's Next?

Black-box Optimization

The Problem

Input

A function $f : \mathscr{X} \to \mathbb{R}$ with domain $\mathscr{X} \subseteq \mathbb{R}^d$

Target

The optimal point $x^* \triangleq \arg\max_{x \in \mathscr{X}} f(x)$.

Black-box Optimization

The Problem

Input

A function $f : \mathcal{X} \to \mathbb{R}$ with domain $\mathcal{X} \subseteq \mathbb{R}^d$

Target

The optimal point $x^* \triangleq \arg\max_{x \in \mathcal{X}} f(x)$.

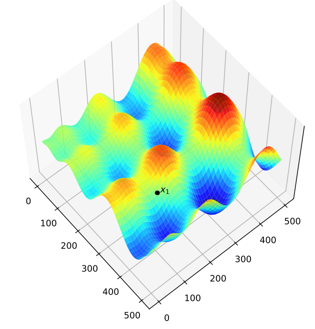
The Iterative Scheme

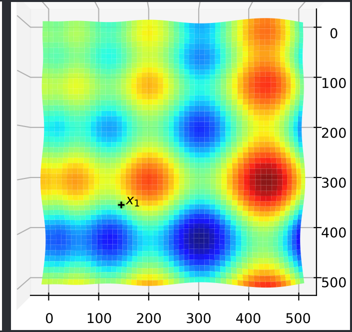
At every iteration $t = 1, 2, \ldots$

1. Choose a point $x_t \in \mathcal{X}$ depending on previous choices of $\{x_i\}_{i=1}^{t-1}$ and evaluations $\{f(x_i)\}_{i=1}^{t-1}$
2. Evaluate the function at x_t and observe $f(x_t)$

Black-box Optimization

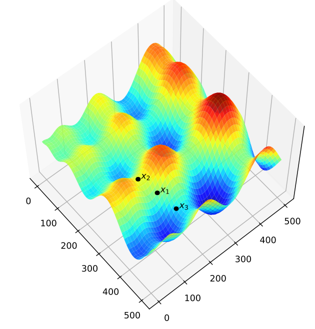
The Hierarchical Tree Search Approach

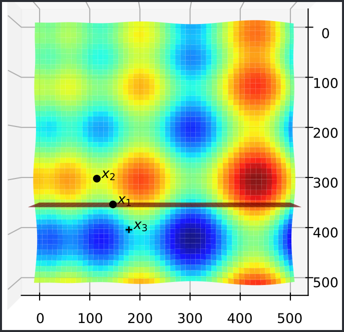


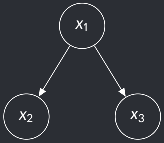


Black-box Optimization

The Hierarchical Tree Search Approach

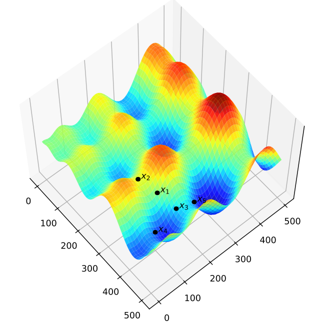


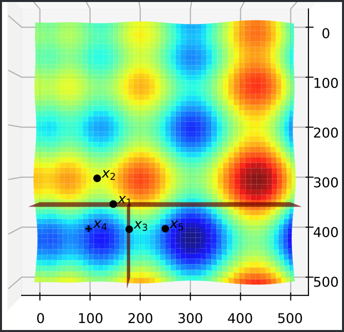


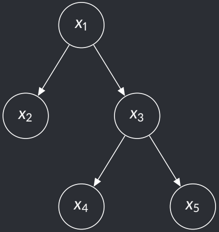


Black-box Optimization

The Hierarchical Tree Search Approach







The Hierarchical Tree Search Framework

How to choose the next point?

Structure: A Tree Cover of $\mathscr{X}$

A tree $\mathscr{T} \subseteq \cup\{(h,l)\}_{h,l=0,1}^{H,2^h}$ of depth H covers the domain $\mathscr{X}$ such that diameter of nodes at level h is bounded by $\nu\rho^h$ for $\rho < 1$.

Construct an Upper Confidence Bound (UCB) to sequentially choose a set of leaf nodes leading to the optimal point

$$
\begin{aligned}
(h_t, l_t) &\triangleq \arg\max_{(h,l)\in\mathscr{T}_t} B^{\min}_{(h,l)}(t) \\
&\triangleq \arg\max_{(h,l)\in\mathscr{T}_t} \min\{B_{(h,l)}(t) + \nu\rho^h, \max_{(h',l')\in\mathrm{Child}(h,l)} B^{\min}_{(h',l')}(t)\}.
\end{aligned}
$$

$B_{(h,l)}(t)$ is the confidence interval constructed around (h,l) at time t.

Theoretical Guarantees

Upper Bounds on Simple Regret: HOO [BMSS11]

Performance Metric: (Expected) Simple Regret

$$\epsilon_T = \mathbb{E}[r_T] = \mathbb{E}[f(x^*) - f(x_T)]$$

Assumption: Weak Lipschitzness of f

$$f^* - f(y) \leq f^* - f(x) + \max\{f^* - f(x), \ell(x, y)\} \forall x, y \in \mathcal{X}$$

HOO: Hierarchical Optimistic Optimization

$$\epsilon_T = O\left(T^{\frac{-1}{d+2}} (\ln T)^{\frac{1}{d+2}}\right)$$

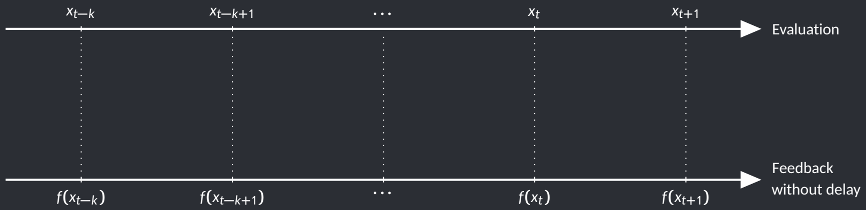
d is the 4ν near-optimality dimension of f.

What's Next?

1. Black-box Optimization: Hierarchical Tree Search
2. Procrastinated Tree Search (PCTS)
    2.1 Adapting to Delayed Feedback
    2.2 Adapting to Noisy Feedback
    2.3 Adapting to Multi-fidelity Feedback
3. Performance Evaluation: Global & Hyperparameter Optimization
4. The Curtain Call: What's Here and What's Next?

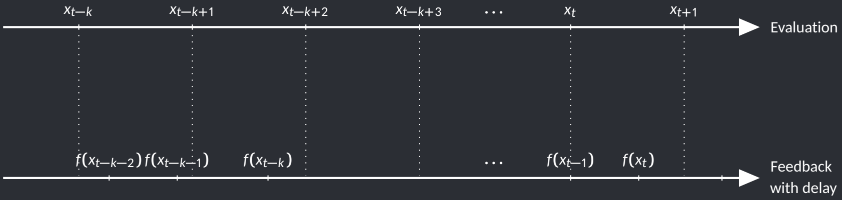
Adapting to Delays

Delays with Bounded Expectations



Adapting to Delays

Delays with Bounded Expectations



Adapting to Delays

Delays with Bounded Expectations

Assumption: Bounded Expectation of Delay

Delays τ_s are generated IID from a distribution D with expectation

$$\tau \triangleq \mathbb{E}[\tau_s : s \geq 0].$$

Adapting to Delays

Adapting Confidence Bounds with Delayed Feedback

PCTS: Adapting to Feedback with Unknown Delays

Generalize upper confidence bound (UCB) of node i from $B_{i,t}$ to $B_{i,s,t}$:

$$s = S_i(t-1),$$

number of evaluation feedbacks from node i observed by time $t-1$.

For any node (h, i), the confidence bound at time t is

$$B_{(h,i),S_{(h,i)}(t-1)}(t) \triangleq \hat{\mu}_{(h,i),S_{(h,i)}(t-1)} + \sqrt{\frac{2\log t}{S_{(h,i)}(t-1)}}$$

Theoretical Guarantees

Upper Bounds on Simple Regret

PCTS: Feedback with Unknown Stochastic and Constant Delays

$$\epsilon_T = O\left(T^{\frac{-1}{d+2}} (\ln T + \tau)^{\frac{1}{d+2}}\right)$$

Wait-and-Act: Feedback with Known Constant Delay

$$\epsilon_T = O\left(T^{\frac{-1}{d+2}} (\tau_{const} \ln T)^{\frac{1}{d+2}}\right)$$

Adaptive strategy is significantly better than waiting (batching).

Theoretical Guarantees

Implications of Theoretical Result

- *Deeper Trees:* The achieved depth of trees grown by PCTS is

$$H \geq \frac{1}{d+2} \frac{\tau + \ln T}{\ln(1/\rho)} = \Omega(\tau + \ln T),$$

  while for wait-and-act HOO $H = \Omega(\ln(T/\tau))$.

- *Benign Delays.* If $\tau = O(\ln T)$, simple regrets of PCTS and HOO are of same order with respect to T.

- *Adversarial Delays.* If $\tau = O(T^{1-\alpha})$ for $\alpha \in (0, 1)$, simple regret

$$\epsilon_T = \tilde{O}(T^{\frac{-\alpha}{d+2}}) = \tilde{O}(\epsilon_T^{HOO} T^{\frac{1-\alpha}{d+2}}).$$

  This echoes the impossibility result of [GVCV20] for finite-arm bandits.

Adapting to Noise

Known and Unknown Noise Variance

$$\boxed{\tilde{f}(X_t)} = \boxed{f(X_t)} + \boxed{\epsilon_t}$$

Noisy Evaluation	True Evaluation	Noise

Adapting to Noise

Setups: Known and Unknown Variance

$$\underbrace{\boxed{\tilde{f}(X_t)}}_{\text{Noisy Evaluation}} = \underbrace{\boxed{f(X_t)}}_{\text{True Evaluation}} + \underbrace{\boxed{\epsilon_t}}_{\text{Noise}}$$

Noise $\leftsquigarrow$ $\mathscr{D}(\mu = 0, \sigma^2)$

Noise Distribution

Adapting to Noise

Designing Confidence Bounds with Noisy Feedback

Settings	Noise-oblivious	Known Variance	Unknown Variance
	DUCB1 [JGS16]	DUCB1σ	
$B_{i,s}(t)$	$\hat{\mu}_{i,s} + \sqrt{\frac{2\log t}{s}}$	$\hat{\mu}_{i,s} + \sqrt{\frac{2\sigma^2 \log t}{s}}$	

Solution: Use the variance σ^2 to calibrate the UCB.

Adapting to Noise

Designing Confidence Bounds with Noisy Feedback

Settings	No Noise	Known Variance	Unknown Variance
	DUCB1 [JGS16]	DUCB1σ	DUCBV
$B_{i,s}(t)$	$\hat{\mu}_{i,s} + \sqrt{\frac{2\log t}{s}}$	$\hat{\mu}_{i,s} + \sqrt{\frac{2\sigma^2\log t}{s}}$	$\hat{\mu}_{i,s} + \sqrt{\frac{2\hat{\sigma}^2_{i,s}\log t}{s}} + \frac{3b\log t}{s}$

Solution: Use the empirical variance with delayed feedback $\hat{\sigma}^2_{i,s}$ and the range of noise $2b$ to calibrate the UCB.

Theoretical Guarantees

Upper Bounds on Simple Regret

PCTS+DUCB1: No Noise

$$\epsilon_T = O\left(T^{-\frac{1}{d+2}}(\ln T + \tau)^{\frac{1}{d+2}}\right)$$

PCTS+DUCB1σ: Known Noise Variance

$$\epsilon_T = O\left(T^{-\frac{1}{d+2}}\left((\sigma/\nu)^2 \ln T + \tau\right)^{\frac{1}{d+2}}\right)$$

PCTS+DUCBV: Unknown Noise Variance

$$\epsilon_T = O\left(T^{-\frac{1}{d+2}}\left(((\sigma/\nu)^2 + 2b/\nu) \ln T + \tau\right)^{\frac{1}{d+2}}\right)$$

The ratios of the variance and the range of noise with respect to the smoothness parameter are the cost of known and unknown noise.

Adapting to Multi-fidelity

Bias and Cost Functions as Quantifiers of Fidelity

$\mathscr{D}(\mu = 0, \sigma^2)$

Noise Distribution

$$\tilde{f}(X_t) = f(X_t) + \epsilon_t + \zeta(Z_h)$$

$\tilde{f}(X_t)$	$f(X_t)$	ϵ_t	$\zeta(Z_h)$
Noisy Evaluation	True Evaluation	Noise	Bias

Adapting to Multi-fidelity

Bias and Cost Functions as Quantifiers of Fidelity

$\mathscr{D}(\mu = 0, \sigma^2)$

Noise Distribution

$$\tilde{f}(X_t) = f(X_t) + \epsilon_t + \zeta(Z_h)$$

Noisy Evaluation

True Evaluation

Noise

Bias

Cost $\lambda(Z_h)$

Cost increases with z_h while bias decreases.

As more computational cost (time/resource) is paid, bias is lessened.

Adapting to Multi-fidelity

Designing Confidence Bounds with Multi-fidelity Feedback

For any node (h, i), the confidence bound at time t is

$$B_{(h,i),S_{(h,i)}(t-1)}(t) \triangleq \underbrace{\hat{\mu}_{(h,i),S_{(h,i)}(t-1)} + \zeta(Z_h)}_{\text{empirical mean of biased evaluations}}$$

$$+ \hat{\sigma}_{(h,i),S_{(h,i)}(t-1)}\sqrt{\frac{2\log t}{S_{(h,i)}(t-1)}} + \frac{3b\log t}{S_{(h,i)}(t-1)}$$

Solution: Use the empirical mean and variance computed from the biased evaluations to calibrate the UCB.

Theoretical Guarantees

Upper Bounds on Simple Regret

PCTS+DUCBV: Unbiased Evaluation

$$\epsilon_T = O\left(T^{-\frac{1}{d+2}}\left(\left((\sigma/\nu)^2 + 2b/\nu\right)\ln T + \tau\right)^{\frac{1}{d+2}}\right)$$

PCTS+DUCBV: Biased Evaluation

$$\epsilon_\Lambda = O\left((H(\Lambda))^{-\frac{1}{d+2}}\left(\left((\sigma/\nu_1)^2 + 2b/\nu_1\right)\ln H(\Lambda) + \tau\right)^{\frac{1}{d+2}}\right).$$

Depth till which the budget of evaluation does not burn out:

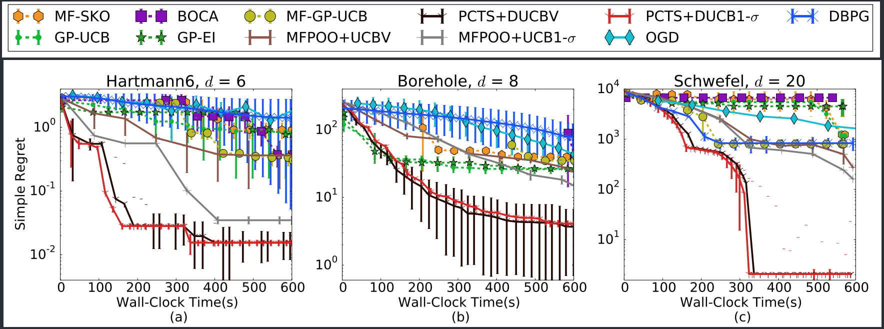
$$H(\Lambda) \triangleq \max\{H : \sum_{h=1}^{H} \lambda(Z_h) \leq \Lambda\}$$

The reduced depth of the tree is the cost of multi-fidelity feedback.

What's Next?

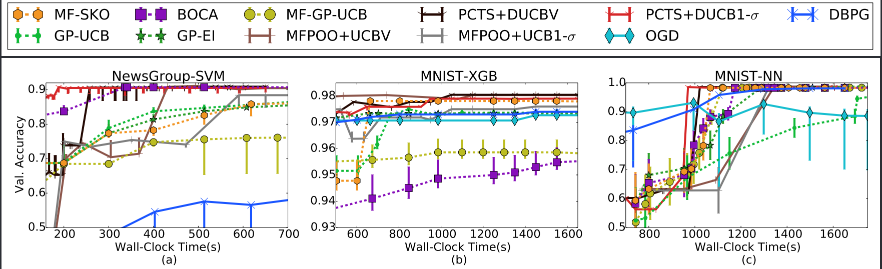
Global Optimization

Benchmark Synthetic Functions



Hyperparameter Optimization

SVM, Random Forest, and Neural Networks



What's Next?

1. Black-box Optimization: Hierarchical Tree Search
2. Procrastinated Tree Search (PCTS)
   2.1 Adapting to Delayed Feedback
   2.2 Adapting to Noisy Feedback
   2.3 Adapting to Multi-fidelity Feedback
3. Performance Evaluation: Global & Hyperparameter Optimization
4. The Curtain Call: What's Here and What's Next?

Summary of Contribution

What's in the Paper?

1. **Algorithmic:** Extending the hierarchical tree search (HOO) framework to PCTS for handling delayed, noisy, and multi-fidelity feedback.

2. **Theoretical:** Incrementally designing upper confidence intervals B that can adapt to delay, noise, and multi-fidelity feedback.

3. **Theoretical:** Developing a unified analysis of regret of PCTS under different feedback settings leading to tighter regret guarantees.

4. **Experimental:** Implementing variants of PCTS on global and hyperparameter optimization benchmarks where PCTS empirically outperforms existing black-box optimizers.

Future Avenues

What's Next?

1. PCTS achieves tighter upper bound than the existing algorithms and the bounds are comparable with the best known upper bounds [CBGMM16] for finite-armed bandits with delays.
   Derive lower bounds for continuum-armed bandits with delayed feedback to understand the fundamental limitations due to delays and tightness of our proposed bounds.
2. In PCTS, we assume to know which delayed feedback corresponds to which evaluation query.
   Extend the framework to anonymously, aggregated delay framework.
3. PCTS solves the continuum-armed bandit problem with delayed, noisy, and multi-fidelity feedback.
   Extend the proposed techniques to solve Markov decision processes with similar feedback.

References I

[BMSS11] Sébastien Bubeck, Rémi Munos, Gilles Stoltz, and Csaba Szepesvári.
X-armed bandits.
Journal of Machine Learning Research, 12(5), 2011.

[CBGMM16] Nicol'o Cesa-Bianchi, Claudio Gentile, Yishay Mansour, and Alberto Minora.
Delay and cooperation in nonstochastic bandits.
In *Conference on Learning Theory*, pages 605–622. PMLR, 2016.

[GVCV20] Manegueu Anne Gael, Claire Vernade, Alexandra Carpentier, and Michal Valko.
Stochastic bandits with arm-dependent delays.
In *International Conference on Machine Learning*, pages 3348–3356. PMLR, 2020.

[JGS16] Pooria Joulani, Andras Gyorgy, and Csaba Szepesvári.
Delay-tolerant online convex optimization: Unified analysis and adaptive-gradient algorithms.
In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.