Unstructured High-Dimensional Bayesian Optimization

Hu Hanyang
Supervisor: Jonathan Scarlett

Department of Mathematics
National University of Singapore
Advanced UROPS in Mathematics for AY2023/2024, Special Term

September 20, 2024

Outline

Introduction

Bayesian optimization (BO) is a popular optimization method for

- black-box (i.e., lacks known special structures)
- expensive-to-evaluate
- noisy

objective functions.

Introduction

Bayesian optimization (BO) is a popular optimization method for

- black-box (i.e., lacks known special structures)
- expensive-to-evaluate
- noisy

objective functions.

With advantages such as:

- Sample-efficiency;
- Uncertainty quantification.

Introduction

Bayesian optimization (BO) is a popular optimization method for

- black-box (i.e., lacks known special structures)
- expensive-to-evaluate
- noisy

objective functions.

With advantages such as:

- Sample-efficiency;
- Uncertainty quantification.

Curse of Dimensionality (CoD). The number of data points required often grows exponentially with the dimensionality.

Introduction

Typical methods for BO to overcome the CoD:

- restrict to local search (e.g., TuRBO, GIBO)
- assume low-dimensional structures (e.g., Add-GP-UCB, REMBO)

which are essentially aimed at reducing the assumed **complexity**.

[1] Hvarfner, Hellsten, and Nardi, *Vanilla Bayesian Optimization Performs Great in High Dimensions*.

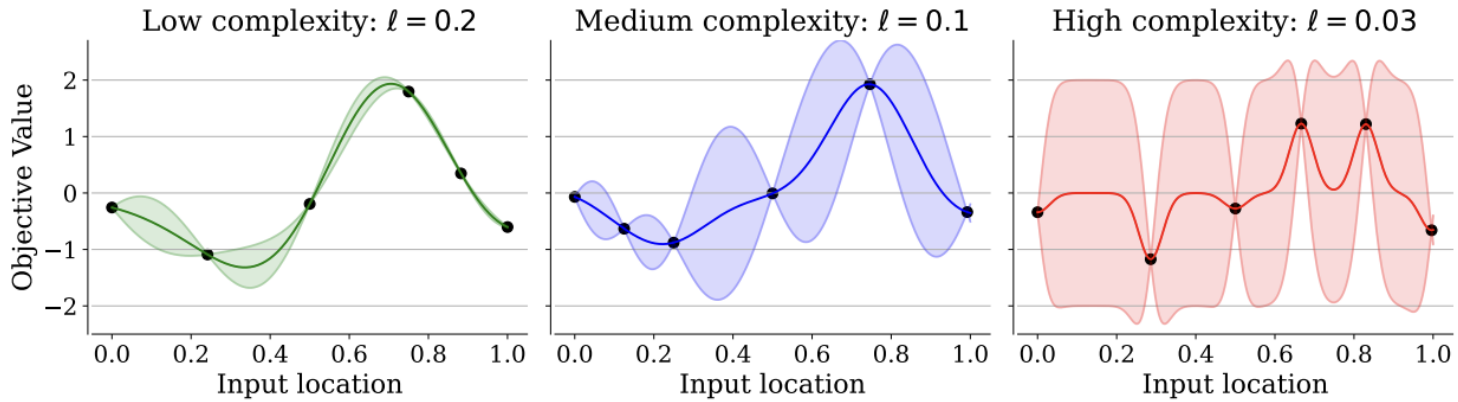
Introduction

Typical methods for BO to overcome the CoD:

- restrict to local search (e.g., TuRBO, GIBO)
- assume low-dimensional structures (e.g., Add-GP-UCB, REMBO)

which are essentially aimed at reducing the assumed **complexity**.

Vanilla BO could be performant in high dimensions **without imposing structural assumptions** - by simply incorporating low-complexity assumptions in the prior of lengthscales.[1]



[1] Hvarfner, Hellsten, and Nardi, *Vanilla Bayesian Optimization Performs Great in High Dimensions*.

Introduction

What are the limitations of this approach?

Are there opportunities to improve upon it?

Background

Algorithm 1 Pseudo Code for Bayesian Optimization

init $\mathbf{X}_0 \subseteq \mathcal{X}$, $\mathbf{y}_0 \subseteq \mathbb{R}$ ▷ generate initial inputs and query observations
$n \leftarrow 1$
for $n \leq N$ **do**
    Fit $(\mathbf{X}_{n-1}, \mathbf{y}_{n-1})$ to obtain a **surrogate model** $\mathcal{M}_n$.
    $\mathbf{x}_n \leftarrow \arg\max_{\mathbf{x} \in \mathcal{X}} \alpha(\mathbf{x} \mid \mathcal{M}_n)$ ▷ optimize **acquisition function**
    $\mathbf{X}_n \leftarrow \mathbf{X}_{n-1} \cup \{\mathbf{x}_n\}$
    $\mathbf{y}_n \leftarrow \mathbf{y}_{n-1} \cup \{f(\mathbf{x}_n) + w_n\}$
    $n \leftarrow n + 1$
end for

Surrogate Model: Gaussian Process

The **Gaussian process (GP)** provides a distribution over functions $\hat{f} \sim \mathcal{GP}(m(\cdot), k(\cdot,\cdot))$ specified by the mean function $m(\cdot)$ and the covariance function $k(\cdot,\cdot)$.

Posterior Inference for Gaussian Process Regression

Assume the observations are corrupted by i.i.d. Gaussian noise $\mathcal{N}(0, \sigma_\varepsilon^2)$ and $m \equiv 0$, then conditioned on $(\mathbf{X}_n, \mathbf{y}_n)$, the predictive distribution of the observation y given input $\mathbf{x}$ is $\mathcal{N}(\mu_n(\mathbf{x}), k_n(\mathbf{x}))$ where

$$\mu_n(\mathbf{x}) = \mathbf{k}_n(\mathbf{x})(\mathbf{K}_n + \sigma_\varepsilon^2 \mathbf{I})^{-1}\mathbf{y}_n$$
$$k_n(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}, \mathbf{x}') - \mathbf{k}_n(\mathbf{x})(\mathbf{K}_n + \sigma_\varepsilon^2 \mathbf{I})^{-1}\mathbf{k}_n^T(\mathbf{x})$$

with $[\mathbf{K}_n]_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$ and $[\mathbf{k}_n(\mathbf{x})]_i = k(\mathbf{x}, \mathbf{x}_i)$ for $i, j \in \{1, \ldots, n\}$.

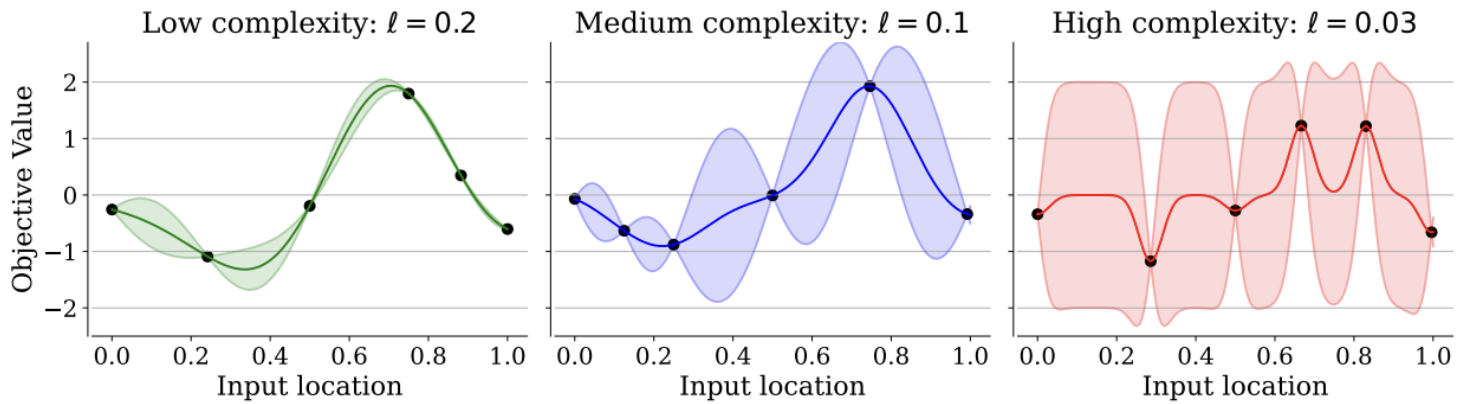
Surrogate Model: Gaussian Process

Radial Basis Function (RBF) Kernel

Let $l \in \mathbb{R}^D_{>0}$ be the lengthscales, the **RBF kernel function** is

$$k(\mathbf{x}, \mathbf{x}') = \exp\left\{-\frac{1}{2}\sum_{i=1}^{d}\frac{(\mathbf{x}_i - \mathbf{x}'_i)^2}{l_i^2}\right\}$$

In this case, the covariance is only dependent on the weighted distance between inputs, i.e. the RBF kernel is **stationary**.



Acquisition Function: Expected Improvement

Expected Improvement (EI)

In the noiseless setting, EI can be computed analytically as

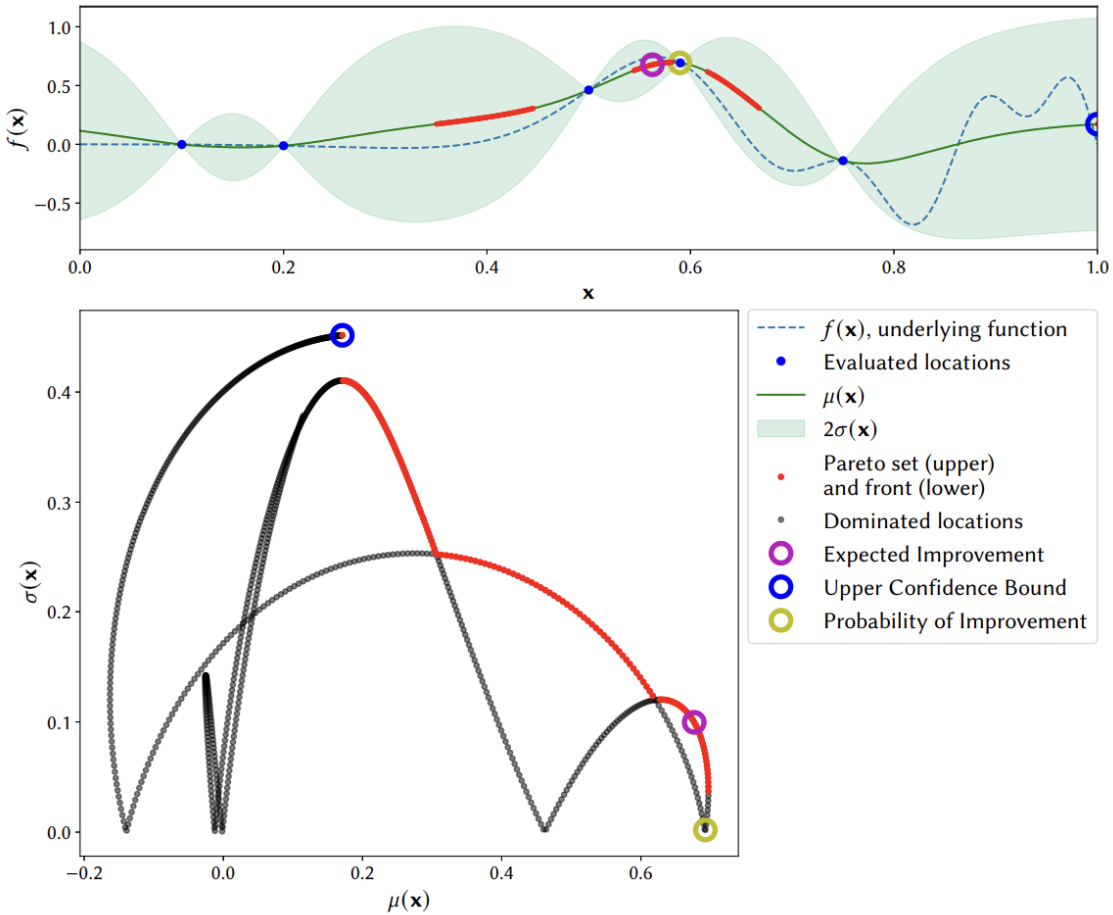
$$\alpha_{\mathrm{EI}}(\mathbf{x} \mid \mathbf{X}_n, \mathbf{y}_n) = \mathbb{E}_{f(\mathbf{x})}\left[[f(\mathbf{x}) - y_{\max}]_+\right] = \sigma_n(\mathbf{x}) h\left(\frac{\mu_n(\mathbf{x}) - y_{\max}}{\sigma_n(\mathbf{x})}\right)$$

where $y_{\max} = \max_{1 \le i \le n} y_i$ is the incumbent and $h(z) = \phi(z) + z\Phi(z)$.

Remark. In the case of noisy observations, we could use Noisy EI (NEI), which is MC-based and differs from EI at the choice of incumbent.

Acquisition Function: Expected Improvement

EI selects a point in the Pareto-optimal set trading-off between exploration and exploitation.[2] See their illustration below:



[2] De Ath et al., "Greed Is Good: Exploration and Exploitation Trade-offs in Bayesian Optimisation".

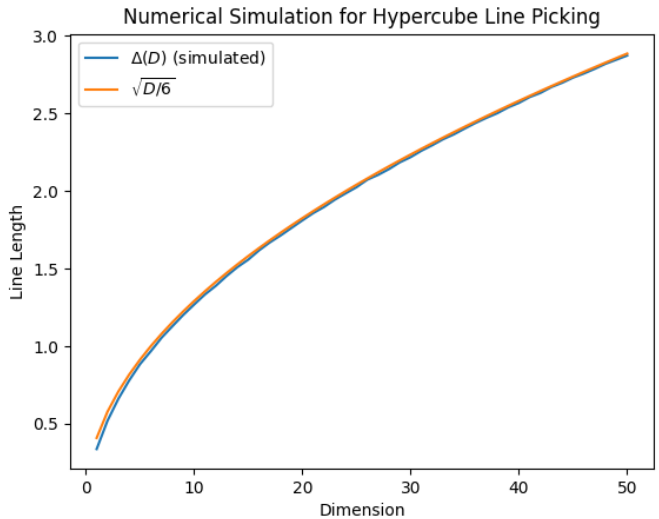
Outline

Revisit CoD for Vanilla BO

Hypercube Line Picking (Anderssen et al., 1976)

The mean distance between uniformly sampled points in a D-dimensional unit hypercube, denoted by $\Delta(D)$, is $\Theta(\sqrt{D})$. More specifically,

$$\frac{1}{3}\sqrt{D} \leq \Delta(D) \leq \sqrt{\frac{D}{6}}\sqrt{\frac{1}{3}\left[1+2\sqrt{1-\frac{3}{5D}}\right]} < \sqrt{\frac{D}{6}}$$



Dimensionality-Scaled Lengthscales Prior

Dimensionality-Scaled Lengthscale Prior (DSP)

To counteract the increase in complexity, Hvarfner et al. propose the DSP:

$$l_n \sim \mathcal{LN}\left(\mu_0 + \frac{\log D}{2}, \sigma_0\right)$$

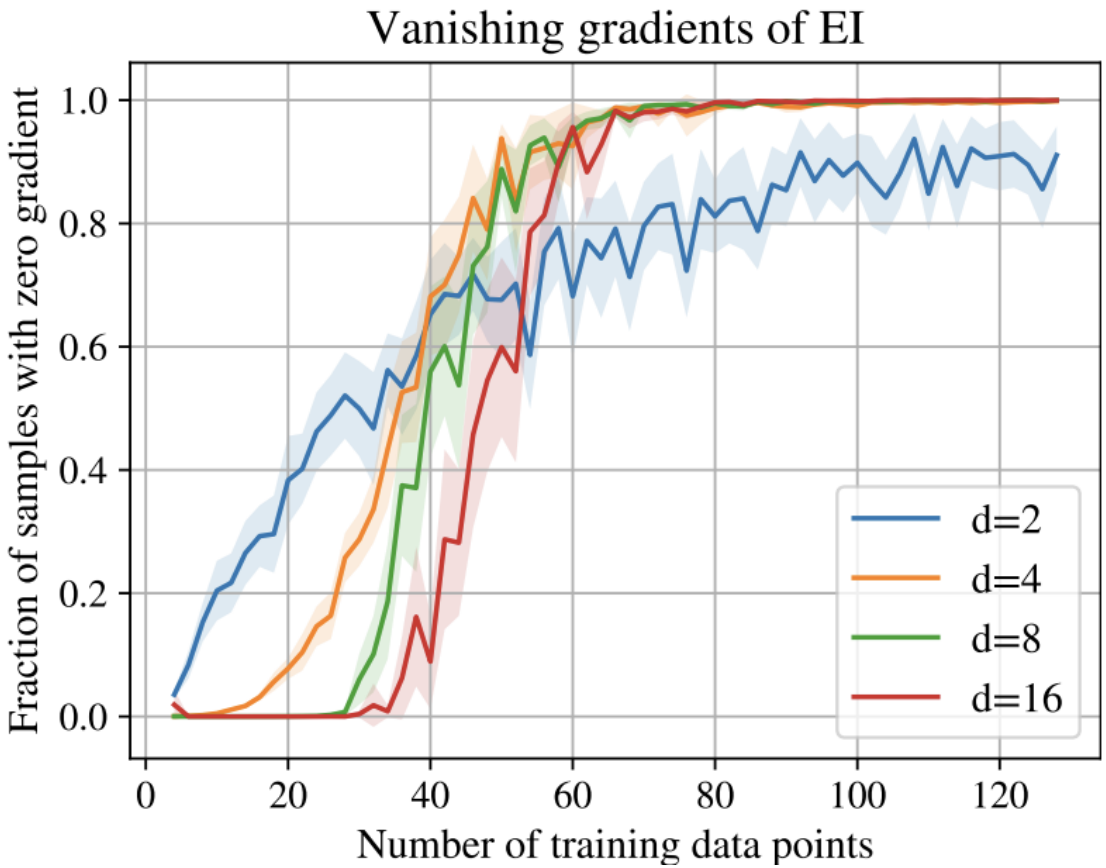
where (μ_0, σ_0) are suitable parameters for a one-dimensional objective.

Remark. The mean and median of the prior scales up by a factor of $\sqrt{D}$.

Remark. They also show that when the model is uninformed, vanilla BO exhibits local search behavior.

Vanishing Gradient

This issue might worsen in high dimensions, take the Ackley function as an example.[3]



[3] Ament et al., *Unexpected Improvements to Expected Improvement for Bayesian Optimization*.

Vanishing Gradient

Ament et al. propose the LogEI family which ensures numerical stability:

- works better than EI when high objective values are concentrated in regions of small volumes;
- yet fails to outperform EI when the model itself is bad.

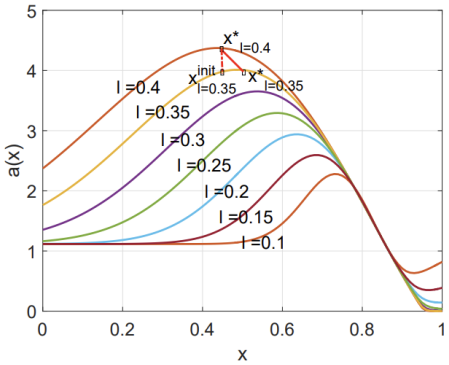
[4] Rana et al., "High Dimensional Bayesian Optimization with Elastic Gaussian Process".

Vanishing Gradient

Ament et al. propose the LogEI family which ensures numerical stability:

- works better than EI when high objective values are concentrated in regions of small volumes;
- yet fails to outperform EI when the model itself is bad.

For high-dimensional settings, Rana et al. propose elastic GP.[4]



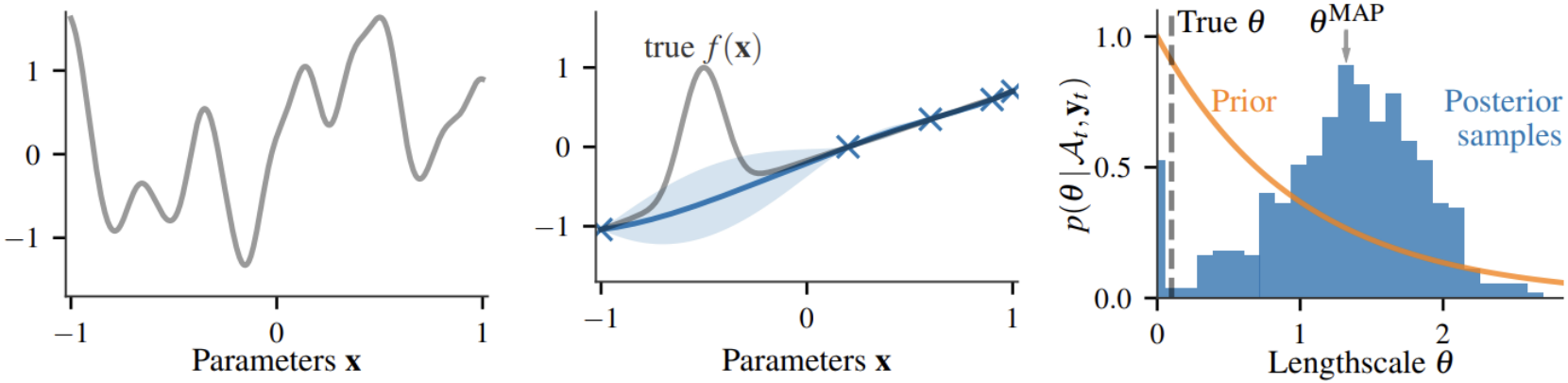
They show that larger lengthscales make acq. function optimization easier.

[4] Rana et al., "High Dimensional Bayesian Optimization with Elastic Gaussian Process".

Unknown Gaussian Process Hyperparameters

The previous two issues seem to support larger lengthscales for BO in high dimensions. However, if lengthscales are erroneously large, the algorithm may miss the global optima.

See the following illustration from Berkenkamp et al.[5]:



(a) Sample from GP prior. (b) GP estimate (RKHS). (c) Lengthscale distribution.

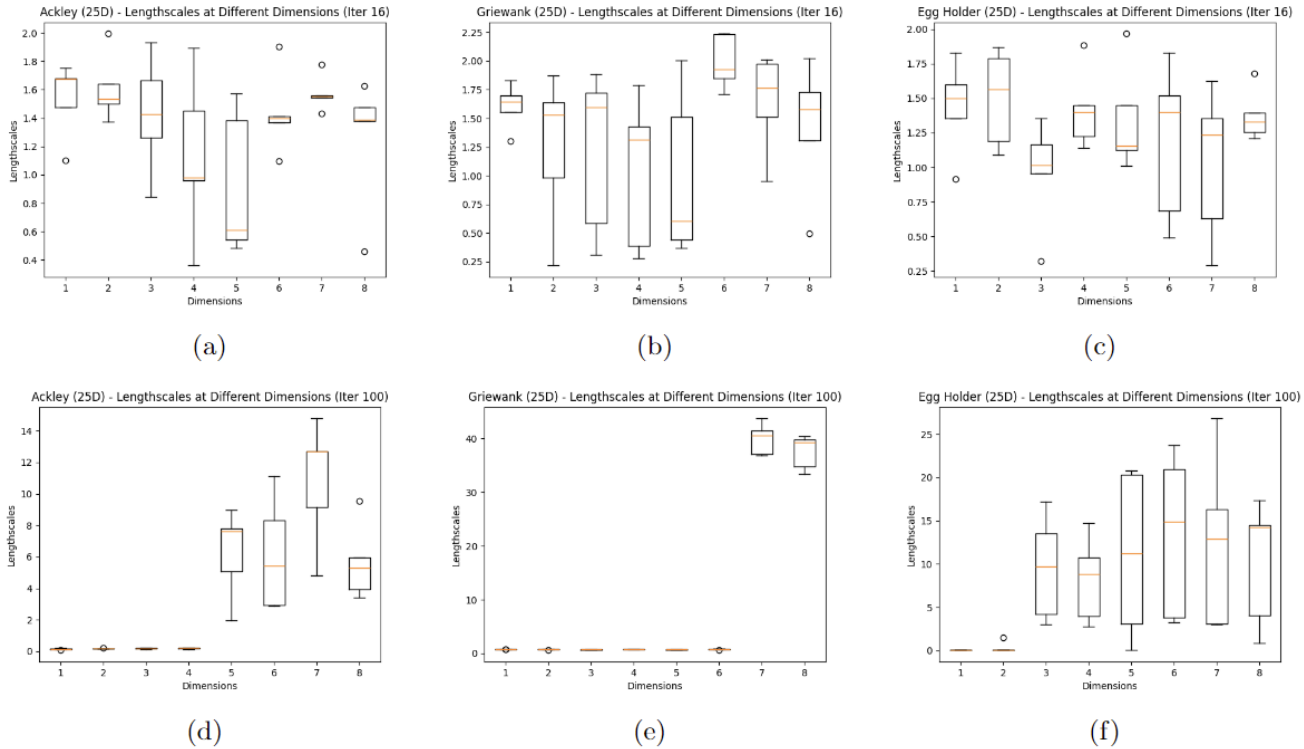
[5] Berkenkamp, Schoellig, and Krause, "No-Regret Bayesian Optimization with Unknown Hyperparameters".

Outline

Lengthscales Cool Down - Meta Strategies

Methods for lengthscales cool down in the low-dimensional setting typically start with an initial guess and shrink them with the proportions being fixed (e.g. AR cool down, A-GP-UCB).

This may not apply to the high-dimensional settings:



Lengthscales Cool Down - Meta Strategies

We consider the following alternatives to shrink the lengthscales:

Option 2 (Shrinking the prior of lengthscales)

Modify the DSP as

$$l_n \sim \mathcal{LN}\left(\mu_0 + \frac{\log D}{2} + \log L, \sigma_0\right)$$

where $L \in [\bar{L}, 1]$ is the base length evolving via certain strategies.

Lengthscales Cool Down - Meta Strategies

We consider the following alternatives to shrink the lengthscales:

Option 2 (Shrinking the prior of lengthscales)

Modify the DSP as

$$l_n \sim \mathcal{LN}\left(\mu_0 + \frac{\log D}{2} + \log L, \sigma_0\right)$$

where $L \in [\bar{L}, 1]$ is the base length evolving via certain strategies.

Option 3 (Shrinking the posterior of lengthscales)

Let l'_n be the lengthscales estimated from the posterior, shrink it via

$$l_n = \max(L l'_n, \bar{l})$$

where L is the same as in option 2 and $\bar{l}$ is a hard lower bound.

Lengthscales Cool Down - Evolution of Base Length

Particular methods for evolving the base length L could be:

- **AR Cool Down** (Wabersich et al.): optimize acquisition function values with the hypothetically shrunk L and the current L respectively, shrink L if the ratio of them is large enough.

Lengthscales Cool Down - Evolution of Base Length

Particular methods for evolving the base length L could be:

- **AR Cool Down** (Wabersich et al.): optimize acquisition function values with the hypothetically shrunk L and the current L respectively, shrink L if the ratio of them is large enough.
- **Fixed Scheduler**: shrinks L according to a pre-defined schedule (potentially depending on the objective's dimensionality);

Lengthscales Cool Down - Evolution of Base Length

Particular methods for evolving the base length L could be:

- **AR Cool Down** (Wabersich et al.): optimize acquisition function values with the hypothetically shrunk L and the current L respectively, shrink L if the ratio of them is large enough.
- **Fixed Scheduler**: shrinks L according to a pre-defined schedule (potentially depending on the objective's dimensionality);
- **Success/Failure Counter**: inspired by TuRBO, expands L after τ_{succ} consecutive successes and shrinks L after τ_{fail} consecutive failures.

Remark. The success/failure counter can potentially recover the preference for larger lengthscales. Furthermore, L tends to shrink when the algorithm consecutively queries points of low posterior variance.

Outline

Adaptive Simplification of Observations

Motivation. Rather than gradually make more complex assumptions, we can try to simplify the observations.

Adaptive Simplification of Observations

Motivation. Rather than gradually make more complex assumptions, we can try to simplify the observations.

What is a "simpler" objective? We could list a few attributes:

- high objective values are not concentrated in small hypervolumes (i.e. the shape is not too spiky);
- less variability from perturbations;
- easier to be modeled by GP (especially with larger lengthscales).

Remark. Multiplying a constant factor $0 < c < 1$ does not help since the observations are standardized.

Adaptive Simplification of Observations

Soft Winsorization

We consider the following transformation of observations:

$$\mathbf{y}' = \sigma_{k,C}((\mathbf{y} - \text{avg}(\mathbf{y}))/\text{std}(\mathbf{y}))$$

where $\sigma_{k,C}(\cdot)$ is a modified sigmoid function defined by

$$\sigma_{k,C}(x) = \frac{x}{\sqrt[k]{1 + \left|\frac{x}{C}\right|^k}}$$

Remark. Notice that

When $k \to \infty$, $\sigma_{k,C}(\cdot)$ approximates **Winsorization**.

When $C \to \infty$, $\sigma_{k,C}(x) \to x$ for all $x \in \mathbb{R}$.

Adaptive Simplification of Observations

Winsorization is a technique used to restrict outliers.

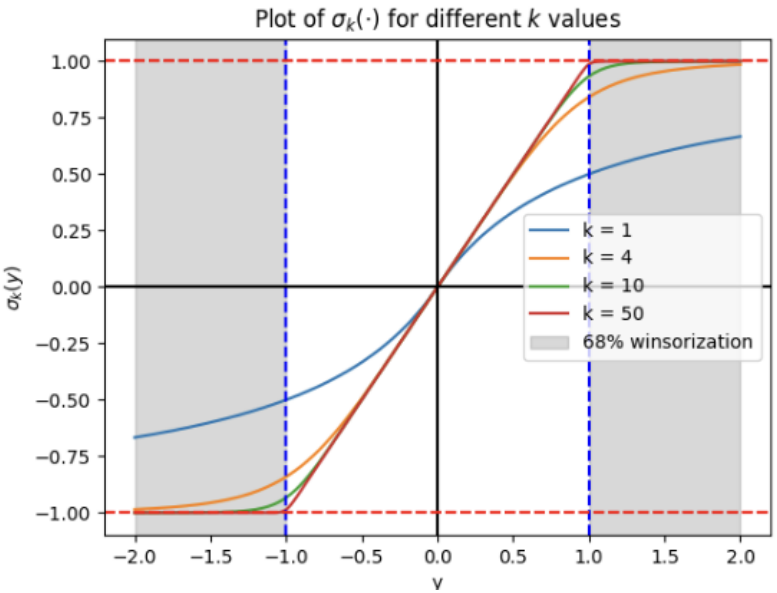
For example, the set $\{-100, -2, -1, 0, 1, 2, 100\}$ would be transformed to $\{-2, -2, -1, 0, 1, 2, 2\}$ after a 68% Winsorization.

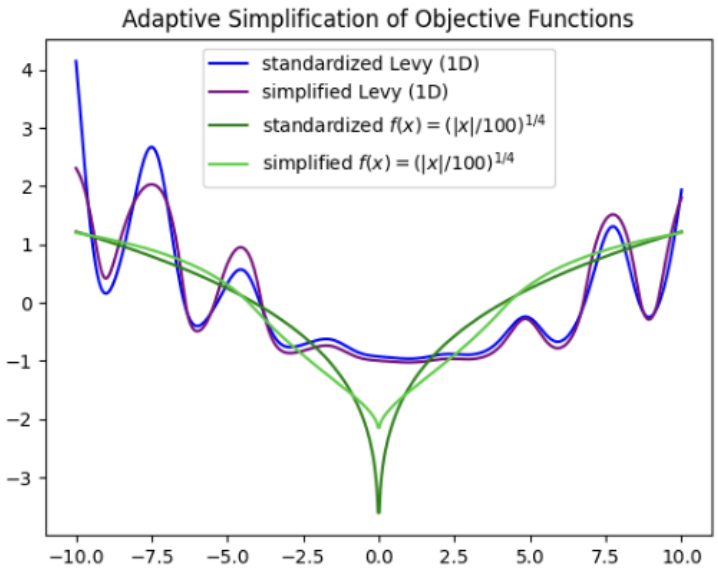
Adaptive Simplification of Observations

Winsorization is a technique used to restrict outliers.

For example, the set $\{-100, -2, -1, 0, 1, 2, 100\}$ would be transformed to $\{-2, -2, -1, 0, 1, 2, 2\}$ after a 68% Winsorization.

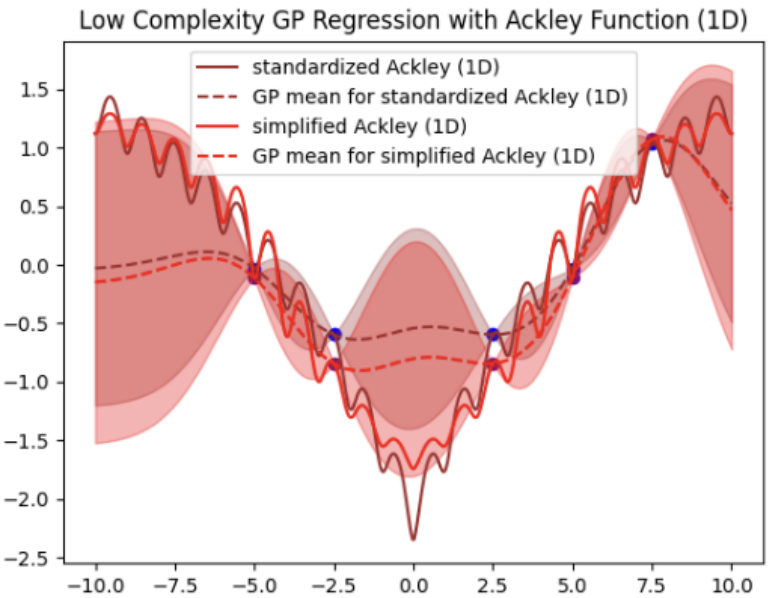
Soft Winsorization could be deemed as a smooth approximation of Winsorization. When $C = 1$, it is corresponding to 68% Winsorization.

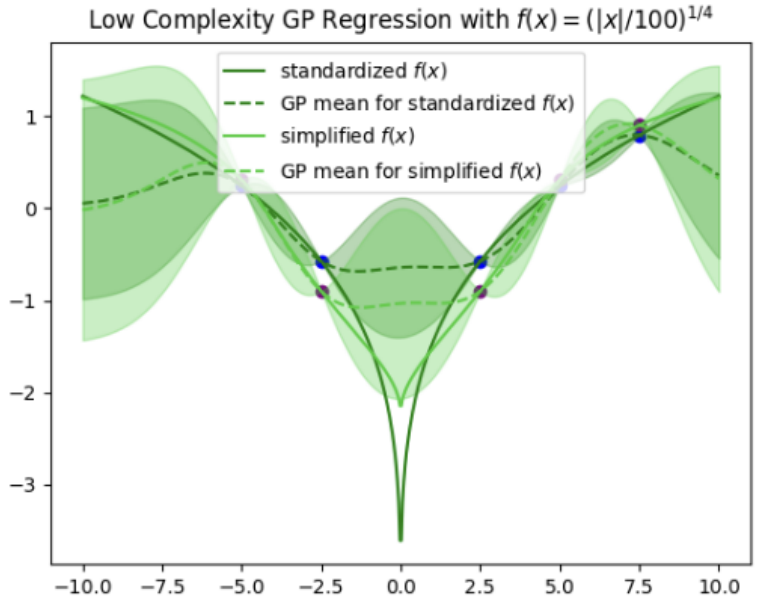




Adaptive Simplification of Observations

For GPs with low (or medium) complexity, objective functions after simplification are deemed more probable.





Adaptive Simplification of Observations

Limitation. Soft Winsorization relies on the estimation of mean and standard deviation. In the context of BO, they might be biased.

Adaptive Simplification of Observations

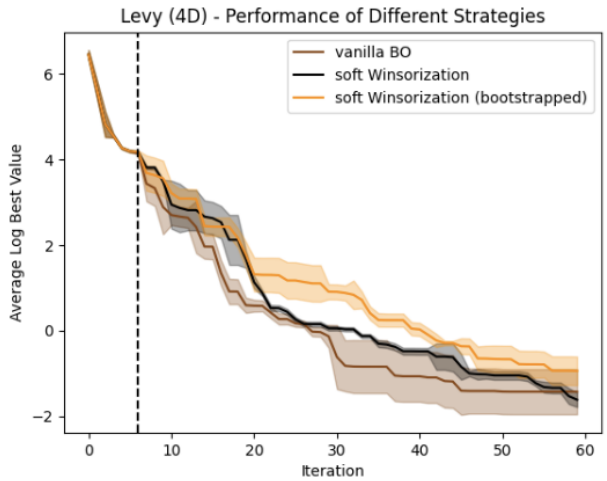
Limitation. Soft Winsorization relies on the estimation of mean and standard deviation. In the context of BO, they might be biased.

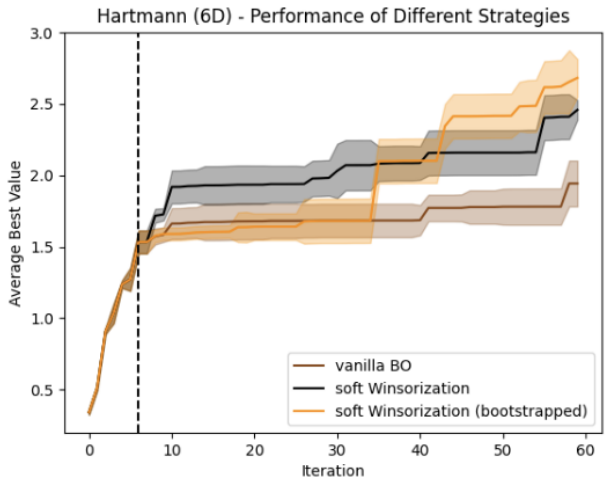
Attempt. Use bootstrapping. Fit a GP on the original observation, sample N points from the posterior, and use their mean and standard deviation.

Remark. This approach has led to performance degradation in most tasks.

Optimization in the Presence of Outliers

Outperform vanilla BO on synthetic tasks (inject i.i.d. Uniform$(-60, 60)$ noise to 16.7% of observations):





More robust for robotics tasks (e.g. learning a 12-D heuristic controller for the lunar lander) that vary a lot for even small perturbations:

	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5
vanilla BO	229.73 ± 27.34	-67.38 ± 97.84	$\mathbf{41.65} \pm 65.33$	37.93 ± 105.82	70.20 ± 145.72
soft Winsorization	$\mathbf{245.46} \pm 24.53$	$\mathbf{18.02} \pm 21.05$	-15.90 ± 40.58	$\mathbf{253.39} \pm 48.16$	$\mathbf{259.26} \pm 17.46$

Outline

Experiment Settings

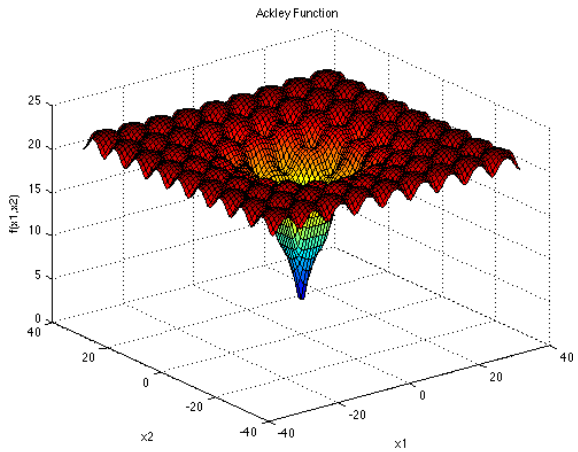
Below are the hyperparameters we used during the experiments:

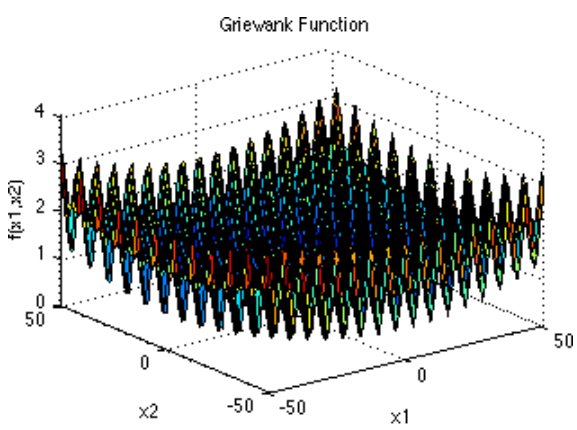
- Vanilla BO (with DSP): $\mu_0 = \sqrt{2}$ and $\sigma_0 = \sqrt{3}$;
- Soft Winsorization (w/wo bootstrapping): $k = 1$ and $C = 1.5$;
- AR Cool Down: thresholding at 1.0 (instead of 1.5);
- Fixed Scheduler: shrink L by 0.7 for every $10\sqrt{D}$ iterations;
- Success/Failure Counter: $\tau_{\text{succ}} = 10$ (instead of 3) and $\tau_{\text{fail}} = \max(4, D)$.

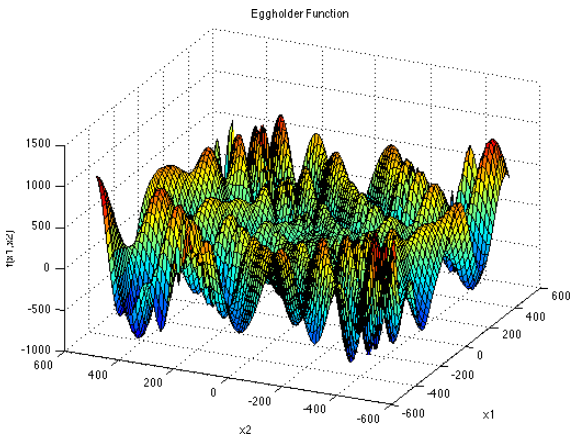
Experiment Settings

For synthetic test functions, we deliberately pick those that are "difficult". E.g., the Ackley function, Griewank function, and egg holder function.

See the following illustrations.[6]







For real-world tasks, we choose Lasso-DNA (180D) and SVM (388D).

[6] Surjanovic and Bingham, *Virtual Library of Simulation Experiments: Test Functions and Datasets*.

Findings

1. Methods implemented using option 2 generally behave more similarly to the vanilla BO compared with their counterpart using option 3.

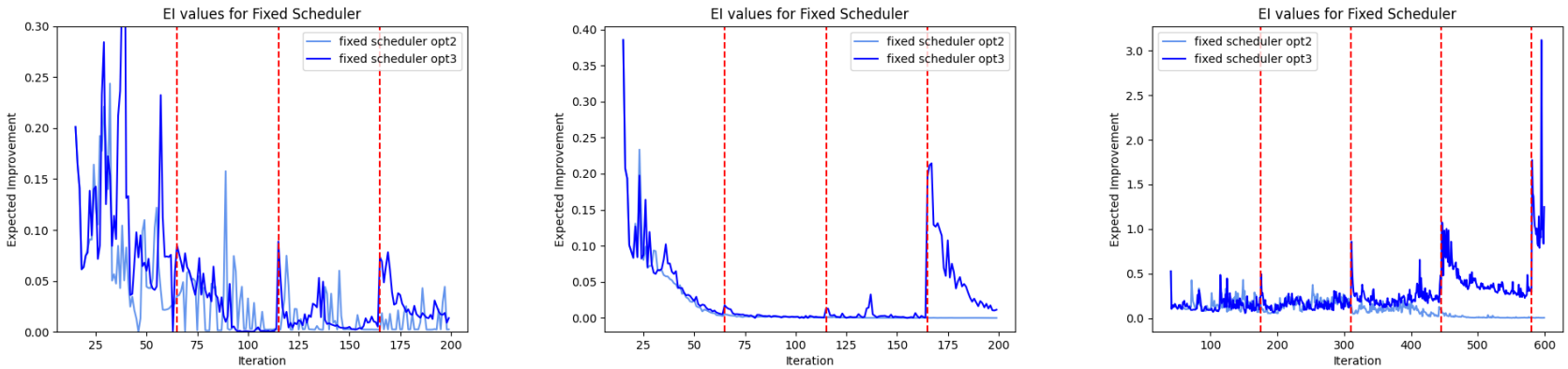
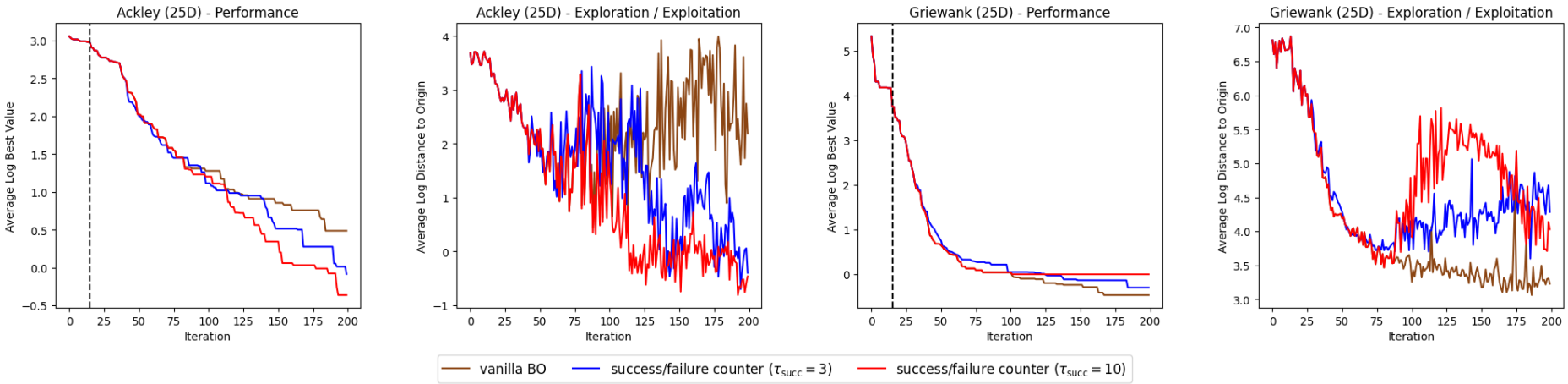


Figure: EI values of the fixed scheduler using options 2 and 3 for the Ackley function (left), the Griewank function (middle), and the Lasso DNA task (right).

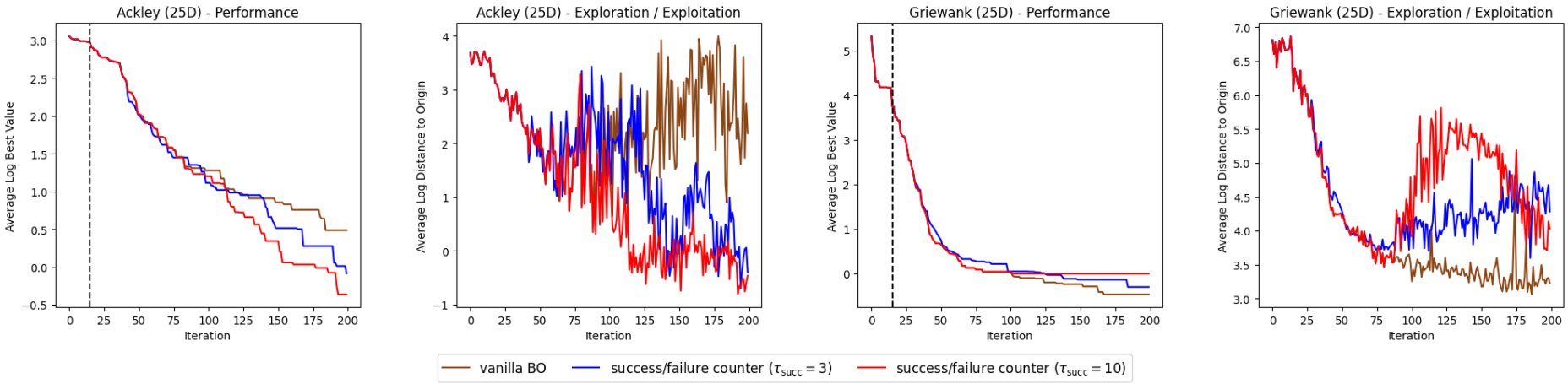
Findings

2. The success/failure counter is able to recover from over-exploration, which might not relate to its potential to recover larger base lengths.



Findings

2. The success/failure counter is able to recover from over-exploration, which might not relate to its potential to recover larger base lengths.



3. No methods consistently outperform the baseline.

Outline

Conclusions and Potential Refinements

In this project, we explored methods attempting to make the model's assumption and the unknown objective function more aligned:

- evolve base length L to scale the prior/posterior of lengthscales;
- simplify observations via soft Winsorization.

Conclusions and Potential Refinements

In this project, we explored methods attempting to make the model's assumption and the unknown objective function more aligned:

- evolve base length L to scale the prior/posterior of lengthscales;
- simplify observations via soft Winsorization.

Here are some possible refinements that could be done:

- more holistic search of good hyperparameters;
- explore ways to shrink the lengthscales non-linearly.