Multi-fidelity Constrained Optimization for Stochastic Black-Box Simulators

Motivation

Challenges faced in real-world optimization:

- High-dimensional parameter space
- Unavailability of gradient (black-box)
- Stochastic simulator
- Constraints
- Computationally expensive simulators

Problem statement

Given:

- scalar valued function $f(\boldsymbol{x}, \boldsymbol{b})$
- a set of constraints $\mathcal{C}(\boldsymbol{x}, \boldsymbol{b}) = \{\mathcal{C}_1(\boldsymbol{x}, \boldsymbol{b}), \ldots, \mathcal{C}_I(\boldsymbol{x}, \boldsymbol{b})\}$

$\boldsymbol{x} \in \mathbb{R}^d$ are the deterministic design variables and $\boldsymbol{b}$ represents a random vector

Robust Optimization formulation:

$$\min_{\boldsymbol{x}} \mathbb{E}_{\boldsymbol{b}}[f(\boldsymbol{x}, \boldsymbol{b})], \quad \text{s.t} \quad \mathbb{E}_{\boldsymbol{b}}[\mathcal{C}_i(\boldsymbol{x}, \boldsymbol{b})] \le 0, \quad \forall i \in \{1, \ldots, I\}$$

To address the above, we introduce Scout-Nd (Stochastic Constrained Optimization for N dimensions)

Constraint Augmentation

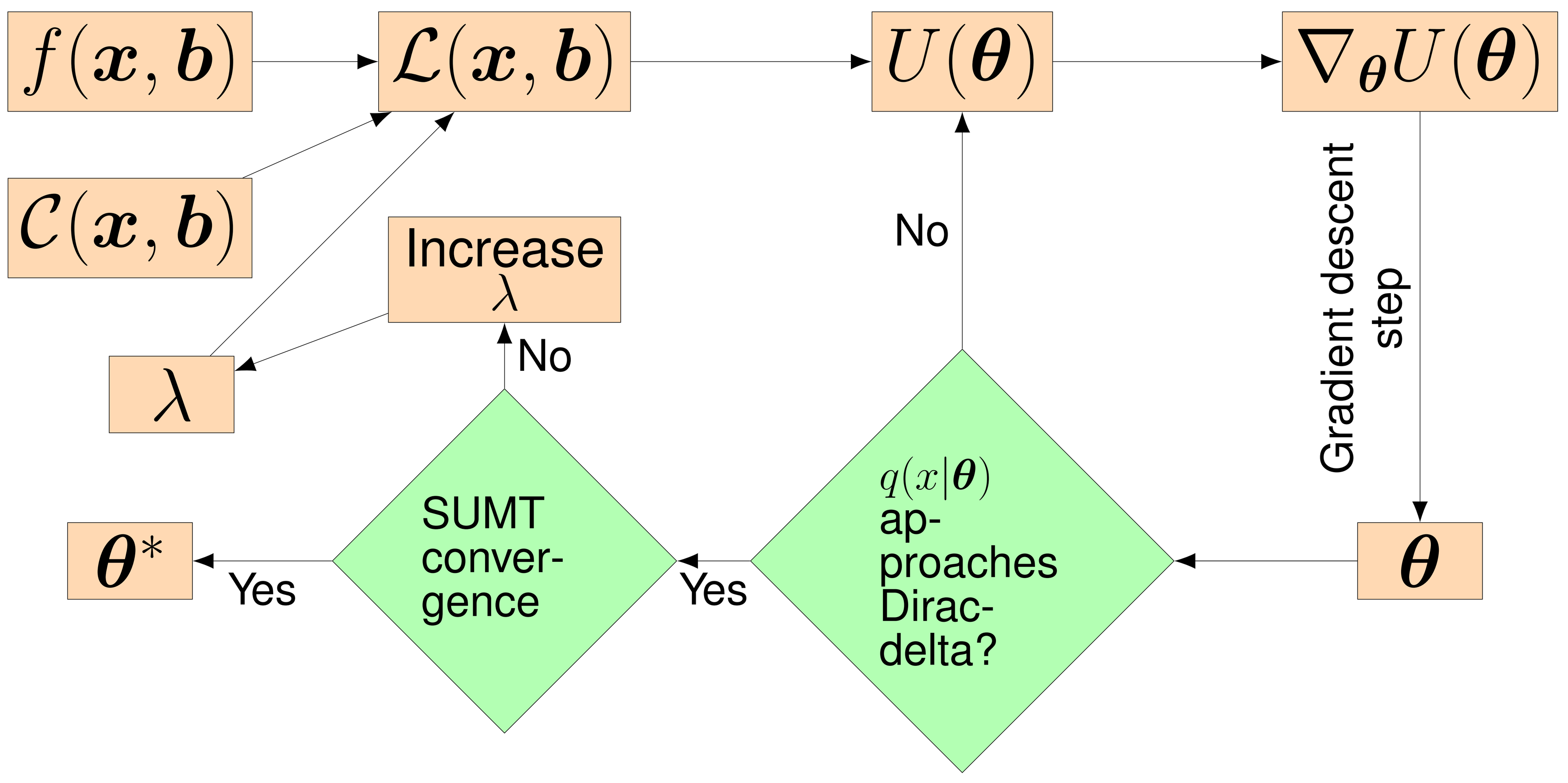
- Cast the constrained optimization to unconstrained optimization.
- Define augmented objective function $\mathcal{L}$

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{b}, \boldsymbol{\lambda}) = f(\boldsymbol{x}, \boldsymbol{b}) + \sum_{i=i}^{I} \lambda_i \max\left(\mathcal{C}_i(\boldsymbol{x}, \boldsymbol{b}), 0\right)$$

- $\lambda_i > 0$ is the penalty parameter for the i^{th} constraint.
- $\max(\cdot, \cdot)$ controls the magnitude of penalty applied.
- The new optimization problem is:

$$\min_{\boldsymbol{x}} \mathbb{E}_{\boldsymbol{b}}[\mathcal{L}(\boldsymbol{x}, \boldsymbol{b}, \boldsymbol{\lambda})]$$

- Start with a small λ and gradually increase its value as per Sequential Unconstrained Minimization Technique (SUMT)[1].



Estimation of derivative

- Minimize the upper bound ($U(\boldsymbol{\theta})$) w.r.t $\boldsymbol{\theta} \iff$ minimize $\mathcal{L}$ w.r.t $\boldsymbol{x}$ [2, 3]

$$\min \int \mathcal{L}(\boldsymbol{x}, \boldsymbol{b}, \boldsymbol{\lambda}) p(\boldsymbol{b}) d\boldsymbol{b} \le \int \mathcal{L}(\boldsymbol{x}, \boldsymbol{b}, \boldsymbol{\lambda}) p(\boldsymbol{b}) q(\boldsymbol{x} \mid \boldsymbol{\theta}) d\boldsymbol{b} d\boldsymbol{x} = U(\boldsymbol{\theta})$$

- $q(\boldsymbol{x} \mid \boldsymbol{\theta})$ is a density over $\boldsymbol{x}$ parameterized by $\boldsymbol{\theta}$. In this work, we only work with gaussian distribution.
- $U(\boldsymbol{\theta})$ converges to $\mathbb{E}_{\boldsymbol{b}}[\mathcal{L}(\boldsymbol{x})]$ when q approaches Dirac-delta.
- Derivative of $U(\boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$ is calculated as:

$$\nabla_{\boldsymbol{\theta}} U(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x}, \boldsymbol{b}}\left[\nabla_{\boldsymbol{\theta}} \log q(\boldsymbol{x} \mid \boldsymbol{\theta}) \mathcal{L}(\boldsymbol{x}, \boldsymbol{b}, \boldsymbol{\lambda})\right]$$

- Estimate the expectation by Monte Carlo

$$\frac{\partial U}{\partial \boldsymbol{\theta}} \approx \frac{1}{S} \sum_{i=1}^{S} \mathcal{L}(\boldsymbol{x}_i, \boldsymbol{b}_i, \boldsymbol{\lambda}) \frac{\partial}{\partial \boldsymbol{\theta}} \log q\left(\boldsymbol{x}_i \mid \boldsymbol{\theta}\right)$$

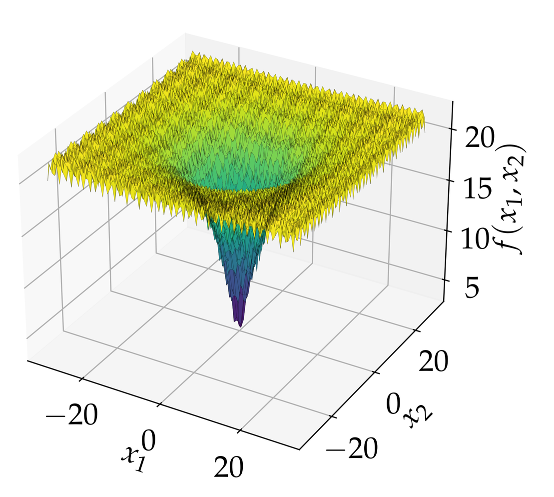
- Variance reduction technique:
  - Baseline method [4]
  - Quasi-Monte Carlo
- Multi-fidelity (MF):
  - Hierarchy of models with increasing accuracy and computational cost.
  - Evaluate the estimation using multi-level Monte Carlo method [5]

$$\frac{\partial U_L}{\partial \theta} \approx \sum_{\ell=1}^{L} \frac{1}{S_\ell} \sum_{i=1}^{S_\ell} \left(\mathcal{L}_\ell(\boldsymbol{x}_i, \boldsymbol{b}_i, \boldsymbol{\lambda}) - \mathcal{L}_{\ell-1}(\boldsymbol{x}_i, \boldsymbol{b}_i, \boldsymbol{\lambda})\right) \frac{\partial}{\partial \boldsymbol{\theta}} \log q\left(\boldsymbol{x}_i \mid \theta\right)$$

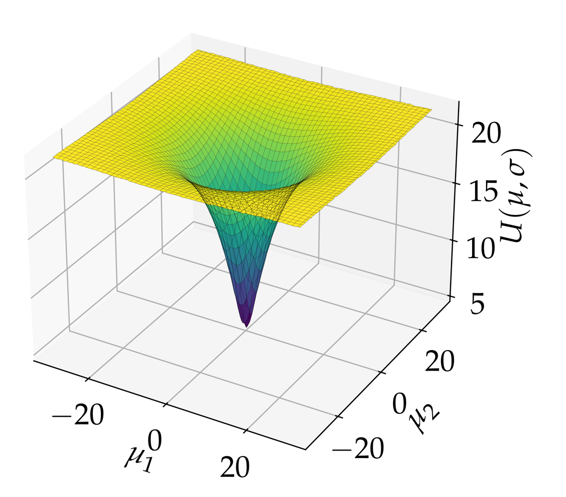
  S_ℓ is the number of samples used at level ℓ and $\mathcal{L}_0(\cdot) = 0$.
  - Less no. of samples allocated to high-fidelity as compared to low-fidelity.
  - This decreases the cost of evaluating expectation when compared to just using high-fidelity function.

Results

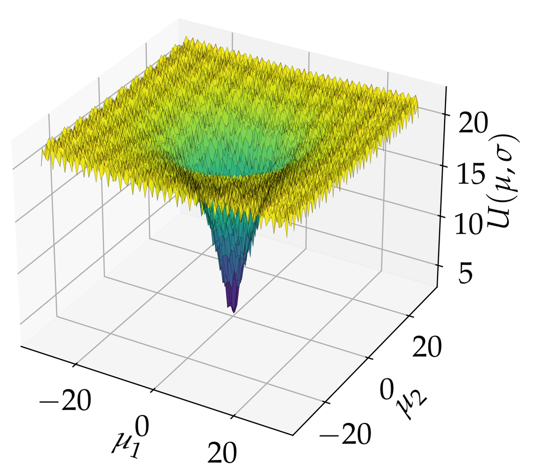
Multi-modal Optimization (Ackley function)



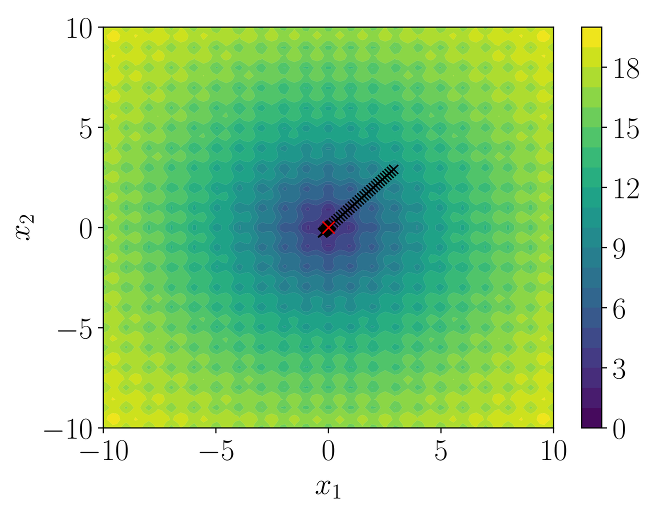
(a) Ackley function



(b) $U(\boldsymbol{\theta})$ for $\sigma^2 = 1$



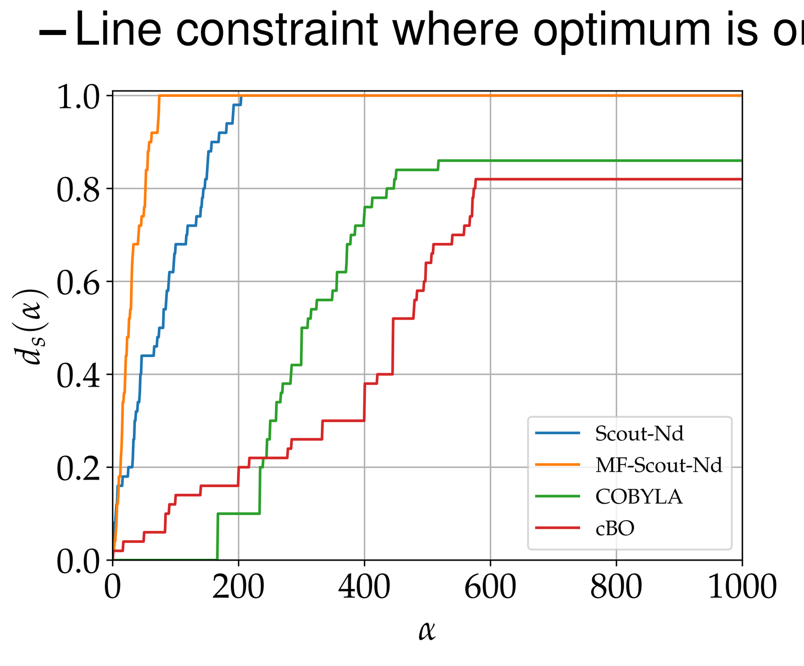
(c) $U(\boldsymbol{\theta})$ for $\sigma^2 = e^{-10}$



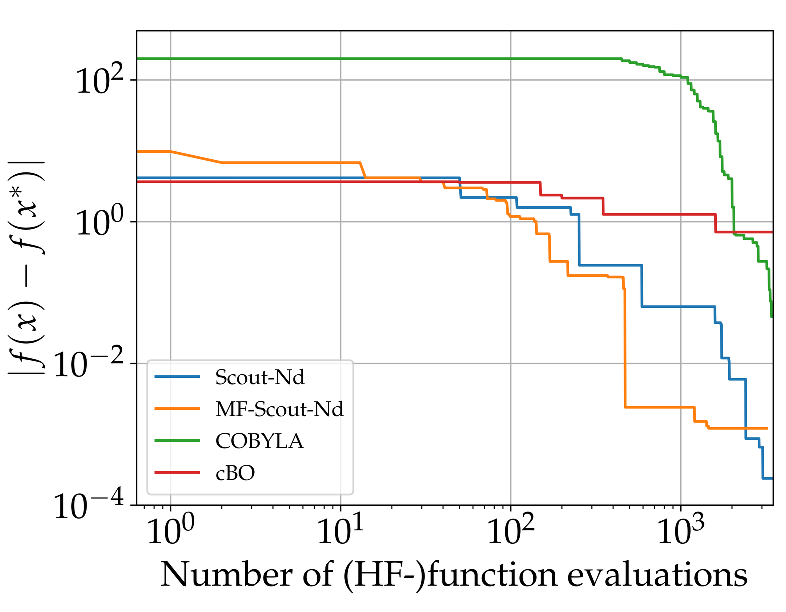
(d) Optimization steps

Benchmarking

- Performance benchmark using data-profile [6] (percentage of problems solved for a given budget, $d(\alpha) = 1.0$: all, $d(\alpha) = 0.0$: none).
- Sphere problem in $2, 4, 8, 16, 32, 64$ dimension, each with:
  - No constraints
  - Line constraint where optimum is on the line



(e) Data Profile



(f) Optimum evolution $d = 8$

- Scout-Nd performs better than cBO and COBYLA because it uses derivative approximation to move toward the optimum which not only helps it to converge faster but also tackled high-dimensionality.
- MF-Scout-Nd performed better as it converged faster towards the optimum than Scout-Nd because it needs fewer costly function evaluations

Atul Agrawal†*, Kislaya Ravi§*, Phaedon-Stelios Koutsourelakis†, Hans-Joachim Bungartz§

{atul.agrawal,kislaya.ravi,p.s.koutsourelakis,bungartz}@tum.de

† Professorship of Data-Driven Materials Modelling, Technical University of Munich, Germany

§ Chair of Scientific Computing, Technical University of Munich, Germany

*equal contribution

Technical University of Munich

NEURAL INFORMATION PROCESSING SYSTEMS

Code, paper and poster:

References

[1] C. Byrne, "Sequential unconstrained minimization algorithms for constrained optimization," *Inverse Problems*, vol. 24, p. 015013, Jan. 2008.

[2] J. Staines and D. Barber, "Variational Optimization," Dec. 2012. arXiv:1212.4507 [cs, stat].

[3] T. Bird, J. Kunze, and D. Barber, "Stochastic Variational Optimization," Sept. 2018. arXiv:1809.04855 [cs, stat].

[4] W. Kool, H. v. Hoof, and M. Welling, "Buy 4 REINFORCE Samples, Get a Baseline for Free!," July 2022.

[5] M. B. Giles, "Multilevel monte carlo methods," *Acta numerica*, vol. 24, pp. 259–328, 2015.

[6] J. J. Moré and S. M. Wild, "Benchmarking derivative-free optimization algorithms," *SIAM Journal on Optimization*, vol. 20, no. 1, pp. 172–191, 2009.