EXPLORING EFFICIENT AND SIMPLE INITIALIZATION STRATEGIES FOR BAYESIAN OPTIMIZATION WITH SETUP-BO

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper investigates the effectiveness of random and grid initialization strategies in the SETUP-BO algorithm which is a self-tuning Bayesian optimization method. We conducted a bunch of experiments on benchmark functions and compare the performance of our suggested initialization strategies to the deterministic initialization strategy. Our results demonstrate that random initialization outperforms other methods, indicating that this method can improve the performance of Bayesian optimization.

1 INTRODUCTION

Optimization methods can be classified into different categories, such as first-order, higher-order, and derivative-free, based on the gradient information they use (Sun et al., 2019). *Bayesian optimization* (BO) (Mockus & Mockus, 1991) is a powerful framework for optimizing unknown or expensive-to-evaluate functions. In this paper, we investigate the effectiveness of two alternative initialization strategies for SETUP-BO[1], a recently proposed self-tuning BO algorithm (Vasconcelos et al., 2022). Specifically, we compare the deterministic initialization strategy to the random initialization and grid initialization strategies, hypothesizing that these simpler strategies can outperform the original for certain types of functions.

In order to evaluate the efficacy of these initialization strategies, we have conducted several experiments on two widely-used benchmark functions in the field of optimization research. Our findings demonstrate that, for these specific benchmarks, the random initialization strategy consistently outperforms alternative methods. This highlights the potential for straightforward strategies to significantly enhance the performance of BO. These results are presented in Section 3.

2 METHODOLOGY

BO relies on the surrogate model to capture the characteristics of the objective function (see A.1) and the acquisition function to suggest the next candidate for assessment (see A.2). To enhance the performance of BO, Auer et al. (1995) introduced *GP-Hedge* algorithm that uses a portfolio of acquisition functions and associates a score to each function based on its performance since the initial step.

However, GP-Hedge has a limitation; it remembers the initial poor performance of each function. Accordingly, if a function has a poor initial performance, its candidate would not have a chance to be selected. To address this limitation, *No-Past-BO* method was introduced which uses a memory decay factor to discount the score over time (Vasconcelos et al., 2019). To further improve the performance, Vasconcelos et al. (2022) introduced SETUP-BO, which auto-tunes the parameter while optimizing the objective function.

Since the initialization of the GP prior to the starting of optimization iterations plays a crucial role in the overall information that BO captures from the objective function, we modified the SETUP-BO

[1]The implementation of SETUP-BO was released https://github.com/thiago-vasconcelos/setup-bo by Vasconcelos et al. (2022)

Table 1: Mann-Whitney U test P-value of SETUP-BO with random initialization against other algorithms. $P \leq 0.05$ indicates that the underlying distribution of the *'absolute error* in each iteration by SETUP-BO with random initialization is significantly lower than the corresponding method.

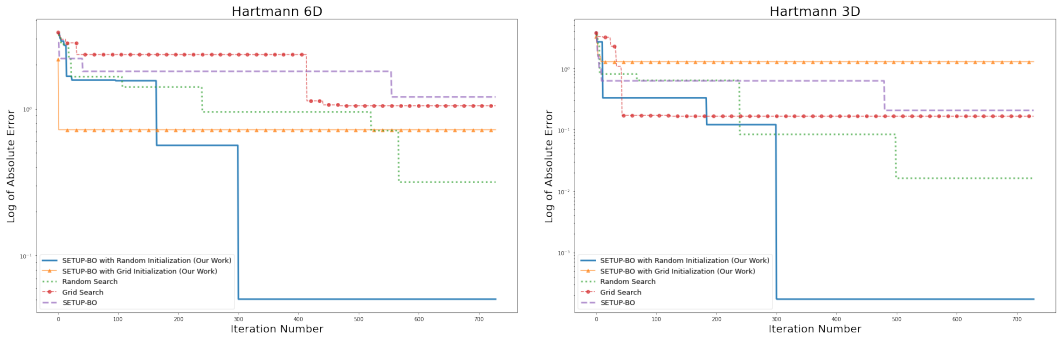
Benchmark	**Random Search**	**Grid Search**	**Deterministic Initialization**	**Grid Initialization**
Hartmann 3D	1.8×10^{-48}	3.8×10^{-73}	7.5×10^{-165}	1.1×10^{-233}
Hartmann 6D	2.6×10^{-69}	3.3×10^{-154}	1.7×10^{-190}	7.6×10^{-76}

implementation to incorporate random and grid initialization strategies. For random initialization, multiple samples drawn from a uniform distribution on the search space of the optimization problem are used to initialize the GP. For grid initialization, the search space is divided into a grid mesh, and the evaluation of the objective function on each corresponding point is used to initialize the GP.

3 Experiments and Results

We compared five different methods including random search, grid search, SETUP-BO with deterministic initialization, random initialization, and grid initialization on the Hartmann 6D and Hartmann 3D benchmark functions, which are 6-dimensional and 3-dimensional uni-modal functions on a unit hyper-cube, respectively. The initialization population is equal for both random and grid initialization techniques.

Figure 1 demonstrates that SETUP-BO with random initialization outperforms other methods on the Hartmann 6D and Hartmann 3D benchmark functions by at least 6 and 93 times more minor errors, respectively. The convergence rate of SETUP-BO with random initialization is also better than other methods on these benchmarks. Notably, SETUP-BO with grid initialization performs poorly, potentially due to converging to local optima resulting from low uncertainty and inability to explore caused by uniform initialization in every region.



(a) (b)

Figure 1: Comparison of *log-absolute error* of the best solution found in each iteration w.r.t. the actual global optima by Random Search, Grid Search, SETUP-BO, SETUP-BO with Grid Initialization, and SETUP-BO with Random Initialization on Hartmann-6D (a) and Hartmann-3D (b)

We also conduct a statistical test using the *Mann-Whitney U* shown in table 1 to verify that SETUP-BO with random initialization has better or equivalent performance compared to other methods.

4 Conclusion

We investigated two simple initialization strategies for SETUP-BO: random and grid initialization. Our experiments on benchmark functions demonstrate that the random initialization strategy outperforms other methods. These findings suggest that this simple strategy can be effective in improving the performance of BO. The ideas of random and grid initialization strategies can also be applied to other population-based optimization methods.

URM STATEMENT

The authors acknowledge that the first author of this work meets multiple of the URM expressed criteria of the ICLR 2023 Tiny Papers Track.

REFERENCES

Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Proceedings of IEEE 36th annual foundations of computer science*, pp. 322–331. IEEE, 1995.

Eric Brochu, Vlad M Cora, and Nando De Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010.

Harold J Kushner. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. 1964.

Jonas Mockus, Vytautas Tiesis, and Antanas Zilinskas. The application of bayesian methods for seeking the extremum. *Towards global optimization*, 2(117-129):2, 1978.

Jonas B Mockus and Linas J Mockus. Bayesian approach to global optimization and application to multiobjective and constrained problems. *Journal of optimization theory and applications*, 70(1): 157–172, 1991.

Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer school on machine learning*, pp. 63–71. Springer, 2004.

Shiliang Sun, Zehui Cao, Han Zhu, and Jing Zhao. A survey of optimization methods from a machine learning perspective. *IEEE transactions on cybernetics*, 50(8):3668–3681, 2019.

Thiago de P Vasconcelos, Daniel ARMA de Souza, César LC Mattos, and João PP Gomes. No-past-bo: Normalized portfolio allocation strategy for bayesian optimization. In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 561–568. IEEE, 2019.

Thiago de P Vasconcelos, Daniel Augusto RMA de Souza, Gustavo C de M Virgolino, César LC Mattos, and João PP Gomes. Self-tuning portfolio-based bayesian optimization. *Expert Systems with Applications*, 188:115847, 2022.

A APPENDIX

As mentioned Bayesian optimization has two major parts: the surrogate model and the acquisition function.

A.1 SURROGATE MODEL IN BAYESIAN OPTIMIZATION

The surrogate model is responsible to capture the characteristics of the objective function and the uncertainty within. The surrogate model is modelled by a Gaussian process. Assume that $f : \mathbb{R}^D \rightarrow \mathbb{R}$ is the function we want to optimize and the evaluation on this function in x_i is:

$$y_i = f(\boldsymbol{x}_i) + \epsilon_i \tag{1}$$

where $\epsilon_i \sim \mathcal{N}(0,\, \sigma^2)$ is a random noise. The Gaussian Process can be written as (Rasmussen, 2004):

$$\begin{aligned} p(\boldsymbol{f}|\boldsymbol{X}) &= \mathcal{N}(\boldsymbol{f}|\boldsymbol{\mu},\, \boldsymbol{K_f}) \\ p(\boldsymbol{y}|\boldsymbol{f}) &= \mathcal{N}(\boldsymbol{y}|\boldsymbol{f},\, \sigma^2\boldsymbol{I}) \\ p(\boldsymbol{y}|\boldsymbol{X}) &= \mathcal{N}(\boldsymbol{y}|\boldsymbol{\mu},\, \boldsymbol{K_f} + \sigma^2\boldsymbol{I}) \end{aligned} \tag{2}$$

where $\boldsymbol{K}_f \in \mathbb{R}^{N\times N}$ is the covariance matrix calculated by the kernel function $k(.,\, .)$ (i.e $[\boldsymbol{K_f}]_{ij} = k(\boldsymbol{x}_i,\, \boldsymbol{x}_j)$).

In inference time, we are interested in predicting the value of function f in $\boldsymbol{x}_*$ which indicates as $\boldsymbol{f}_*$:

$$\begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{f}_* \end{bmatrix} = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\mu}_* \end{bmatrix}, \begin{bmatrix} \boldsymbol{K} & \boldsymbol{K}_* \\ \boldsymbol{K}_*^T & \boldsymbol{K}_{**} \end{bmatrix} \right) \tag{3}$$

where $[\boldsymbol{\mu}]_i$ is the mean for $\boldsymbol{x}_i$. $\boldsymbol{K}$, $\boldsymbol{K}_*$ and $\boldsymbol{K}_{**}$ are the covariance matrix of train-train, train-test and test-test respectively.

A.2 Acquisition Function in Bayesian Optimization

The main goal of the acquisition function is to suggest the next point to evaluate considering the captured data by the surrogate model (e.g. uncertainty of estimation in each point). There are 3 most used acquisition functions (Vasconcelos et al., 2019):

- *Probability of Improvement* that consider the probability for each point being better that the best found optimum (Kushner, 1964).

$$PI(\boldsymbol{x}) = p\{f(\boldsymbol{x}) \leq \boldsymbol{\mu}^- - \xi\} = \Phi\left(\frac{\boldsymbol{\mu}^- - \xi - \mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})}\right) \tag{4}$$

  where $\boldsymbol{\mu}^- = min_i \mu(\boldsymbol{x}_i)$ and $\mu(\boldsymbol{x})$ and $\sigma(\boldsymbol{x})$ are the estimated mean and standard deviation in $\boldsymbol{x}$ respectively. Φ also indicates the CDF of standard Gaussian distribution. ξ adjusts the balance between exploration and exploitation.

- *Expected improvement* that also take the amount of improvement into account (Mockus et al., 1978).

$$EI(\boldsymbol{x}) = \mathbb{E}[\boldsymbol{\mu}^- - \mu(\boldsymbol{x})] = \tau(\boldsymbol{x})\Phi\left(\frac{\tau(\boldsymbol{x})}{\sigma(\boldsymbol{x})}\right) + \sigma(\boldsymbol{x})\phi\left(\frac{\tau(\boldsymbol{x})}{\sigma(\boldsymbol{x})}\right) \tag{5}$$

  where $\tau(\boldsymbol{x}) = \boldsymbol{\mu}^- - \xi - \mu(\boldsymbol{x})$ and ϕ is the PDF of standard Gaussian distribution.

- *GP - Lower confidence bound* (or GP - Upper confidence bound in maximization problems) that use a linear combination of the mean and standard deviation of the GP (Brochu et al., 2010).

$$GP - LCB(\boldsymbol{x}) = \mu(\boldsymbol{x}) - k\sigma(\boldsymbol{x}) \tag{6}$$

  where k adjust the balance between exploration and exploitation

Figure 2 and figure 3[2] show two consecutive iterations of BO with GP-UCB demonstrating how the surrogate model and acquisition function work to suggest the next candidate for evaluation.

[2]These figures were adapted from https://github.com/fmfn/BayesianOptimization/blob/master/examples/visualization.ipynb.

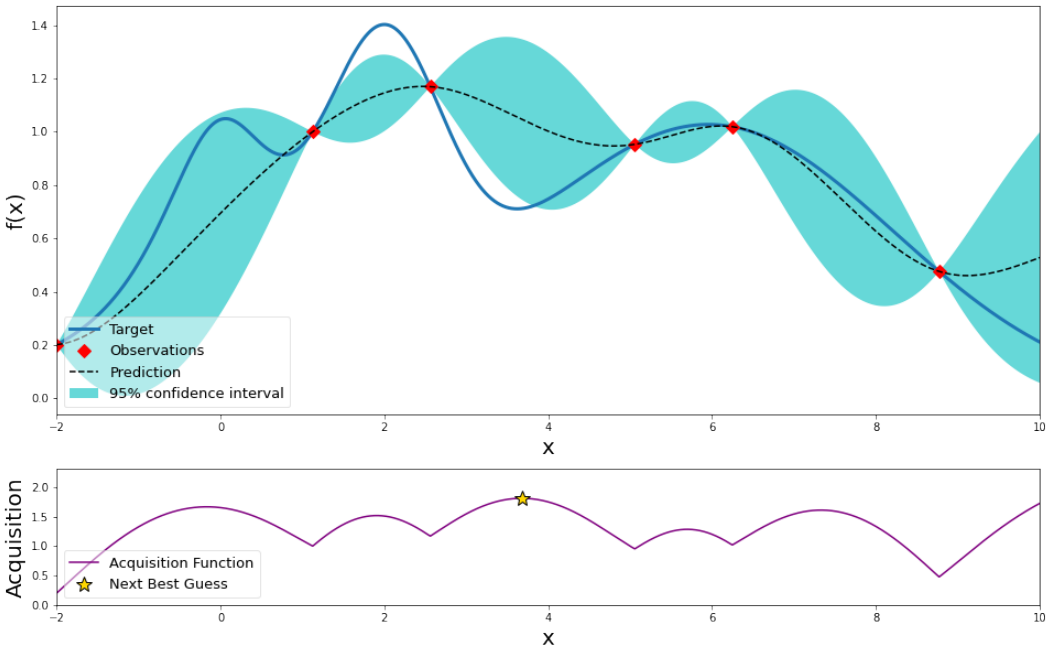


Figure 2: Top: The Bayesian Optimization in the 5th iteration; the blue graph and the dotted black one show the objective function and the prediction of the objective function (a.k.a. surrogate model) respectively. The red dots indicate the observation, and the cyan region shows the uncertainty with a confidence level of 95%
Bottom: The acquisition function (GP-UCB) and the star shows the next candidate for evaluation

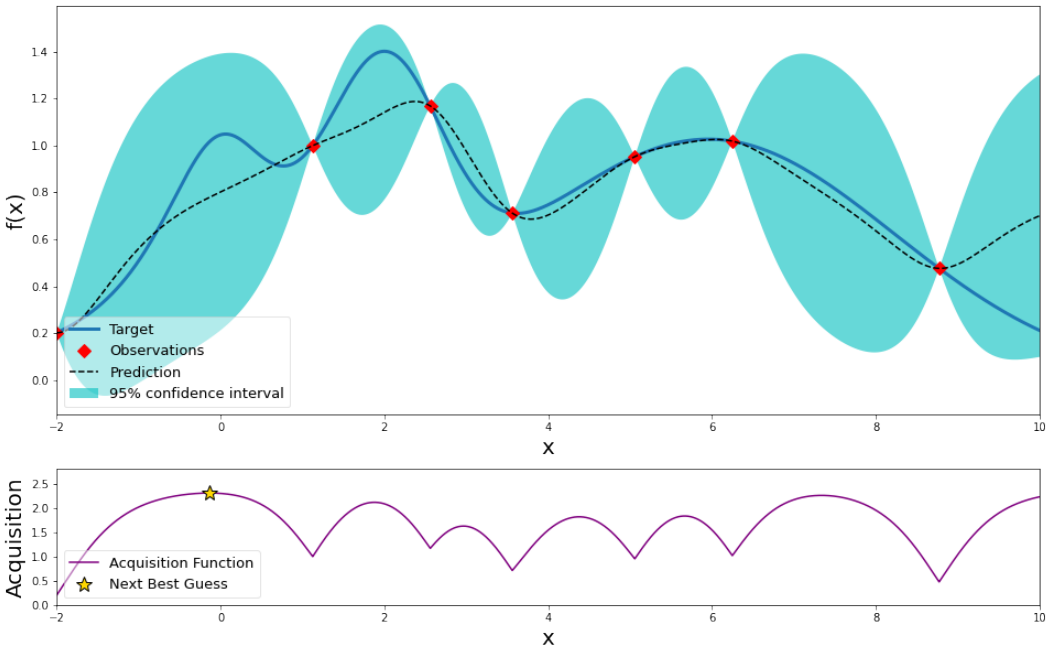


Figure 3: Top: The Bayesian Optimization in the 6th iteration; As demonstrated, the uncertainty around the evaluated point is decreased.
Bottom: The acquisition function (GP-UCB) and the star shows the next candidate for evaluation.