High Dimensional Optimization for Electronic Design

Yuejiang Wen Jacob Dean Brian A. Floyd Paul D. Franzon

{wyuejia,jtdean2,bafloyd,paulf}@ncsu.edu Department of Electrical and Computer Engineering North Carolina State University Raleigh, North Carolina, USA

ABSTRACT

Bayesian optimization (BO) samples points of interest to update a surrogate model for a blackbox function. This makes it a powerful technique to optimize electronic designs which have unknown objective functions and demand high computational cost of simulation. Unfortunately, Bayesian optimization suffers from scalability issues, e.g., it can perform well in problems up to 20 dimensions. This paper addresses the curse of dimensionality and proposes an algorithm entitled Inspection-based Combo Random Embedding Bayesian Optimization (IC-REMBO). IC-REMBO improves the effectiveness and efficiency of the Random EMbedding Bayesian Optimization (REMBO) approach, which is a state-of-the-art high dimensional optimization method. Generally, it inspects the space near local optima to explore more points near local optima, so that it mitigates the over-exploration on boundaries and embedding distortion in REMBO. Consequently, it helps escape from local optima and provides a family of feasible solutions when inspecting near global optimum within a limited number of iterations.

The effectiveness and efficiency of the proposed algorithm are compared with the state-of-the-art REMBO when optimizing a mmWave receiver with 38 calibration parameters to meet 4 objectives. The optimization results are close to that of a human expert. To the best of our knowledge, this is the first time applying REMBO or inspection method to electronic design.

CCS CONCEPTS

- Hardware → Electronic design automation; Integrated circuits; Wireless devices; Emerging tools and methodologies; • Software and its engineering \rightarrow Search-based software engineering;
- Computing methodologies → Bayesian network models; Factorization methods; • Mathematics of computing → Bayesian computation; Solvers.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2022 Association for Computing Machinery. ACM ISBN 978-1-4503-9486-4/22/09...\$15.00 https://doi.org/10.1145/3551901.3556495

MLCAD '22, September 12-13, 2022, Snowbird, UT, USA

KEYWORDS

Electronic Design Automation (EDA), High Dimensions, Bayesian Optimization, Random Embeddings, Local Inspection, Analog Cir-

ACM Reference Format:

Yuejiang Wen, Jacob Dean, Brian A. Floyd, and Paul D. Franzon. 2022. High Dimensional Optimization for Electronic Design. In Proceedings of the 2022 ACM/IEEE Workshop on Machine Learning for CAD (MLCAD '22), September 12-13, 2022, Snowbird, UT, USA. ACM, New York, NY, USA, 5 pages. https:// //doi.org/10.1145/3551901.3556495

1 INTRODUCTION

Bayesian optimization (BO) is a sample-efficient method for expensive unknown function optimization, which makes it a useful method in many ubiquitous problems such as circuit design optimization [9, 19, 21], machine learning [1, 17], computer graphics and visual design[2]. Bayesian optimization consists of two parts: acquisition function and a surrogate model (e.g., Gaussian process(GP)). The framework of BO consists of three iterative phases: 1) Based on beliefs about the behavior of the objective function, the acquisition function[17](e.g., maximum probability of improvement (MPI), expected improvement (EI) and upper confidence bound (UCB)) samples the point of interest; 2) Send sample to evaluator (objective function, f) to get feedback; 3) The surrogate model augments the [sample, feedback] and updates the model for sampling next sample of interest iteratively until reaching maximum number of iterations.

Although BO is a powerful method, it suffers from scalability issues[5, 8, 22], e.g., it performs well in problems up to 10-20 variables[12, 13]. As the number of optimization dimension increases, the computational complexity of covariance matrix inversion (the precision matrix) and the evaluation of the acquisition function at different points increase exponentially[16].

The improvements in Bayesian optimization have been demonstrated solving higher dimensional problems, including rerepresenting original dataset using low intrinsic dimensions (random embedding BO (Fig. 1) [14, 22], Subspace Identification Bayesian Optimization (SI-BO) [5], Sparse BO (or sparse Gaussian processes (SGPs)[15]), Sliced Inverse Regression Bayesian Optimization(SIRBO) [23], Hashing enhanced Subspace BO (HeSBO)[13], Non linear embeddingmethod [11]), objective function decomposition[8], dropout strategy[7]. Recently, BO has been applied to high dimensional analog design optimization. In [20], authors use an ensemble/bagging method for analog sizing. The proposed approach uses an ensemble of GP models to approximate the objective and constraint functions

locally in the search space. Unlike mainstream BO approaches, this method is able to traverse high dimensional problems with ease and provide multiple query points for parallel evaluation. However, it may not be feasible for high-correlation/ nonconvex problems and needs to query many points for a number of GP models, which can be computationally expensive. Similar work has also been done in [18]. Authors proposed Bayesian Optimization with Deep Partitioning Tree (DPT-BO) for high-frequency design optimization. However, it needs additive process and components. Among these

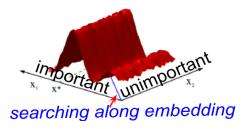


Figure 1: Random EMbedding Bayesian Optimization (REMBO): This 2-D function only has 1 important dimension. After embedding 2-D to 1-D space (the blue embedding line), it would be more efficient to search for the optimum along the 1-D embedding than in the original 2-D space [22].

methods, Random EMbedding Bayesian Optimization (REMBO) does not need any sensitivity analysis steps or additive structures. [22] demonstrates REMBO on non-circuit problems with up to 47 dimensions. The key idea in REMBO is that even large dimensional problems can be remapped to fewer effective dimensions and then solved directly (Fig.1)[22]. These effective dimensions can include combinations of the original dimensions. These reduced order effective dimensions are discovered by trying multiple random embeddings (i.e. mappings of the higher dimensional space to the low dimensional space). REMBO solves three problems. First, it eliminates the need for humans to do a sensitivity analysis and eliminate dimensions manually. Such human steps require detailed knowledge of both the algorithm and circuit, a rare combination. Second, REMBO is not restricted to cases where the final axes for the dimensions are aligned with those in the original problem. For example, there might be a problem where the objective function is not sensitive to R1 and R2, as long as R1 and R2 track each other. REMBO will discover that, and eliminate one original dimension through dimensional remapping, e.g., random matrix (2×1) · sample $(y_1) = \binom{R_1}{R_2}$, so that the 2-D problem can be converted to a 1-D problem. REMBO, therefor, has potential to discover dimensional remappings through recombinations that human guided sensitivity analysis could not. Third, REMBO has theoretical guarantees as to the probability of success[22].

Fig. 2 shows the work flow of using optimizer, e.g., BO or REMBO to find the best design (including transistor sizes, inductance, capacitance, etc.) for a given analog circuit topology. The optimizer can also be applied to different IPs by switching the available sampling strategies and surrogate models in optimizer, e.g., random forest, Gaussian process model. First, the optimizer samples a sample vector, X_{sample} based on sampling strategies. Then, X_{sample} is multiplied by a random matrix A to project X_{sample} to a high

dimension space that aligned with true dimensions, i.e., number of tuning parameters of the circuit. After this random embedding process, the embedded vector is used to modify a pre-generated ocean file. The modified ocean file is sent to the simulator to get a weighted value of multi-performance indicators for optimizer to update the sampling function and sample the next interested sample, iteratively. After it reaches the maximum number of iterations of optimization, optimizer provides the best solution.

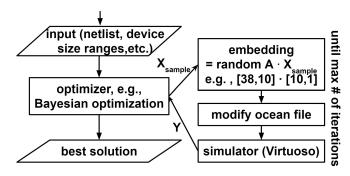


Figure 2: The Process of Applying REMBO on a Circuit Design.

Although REMBO is a simple method to optimize high dimensional problems, its efficiency is low due to both embedding distortion and the over-exploration of boundary issues. Since in high dimensions data points typically lie mostly on the boundary, and anyways far away from each other, the predictive variance tends to be higher in the regions near the boundary. This is a waste of computation effort assuming the global minimum is not on the boundary[10]. The global optimum may fall outside of the bound of box. Sometimes REMBO needs to try different random embeddings.

2 THE PROPOSED INSPECTION-BASED COMBO REMBO (IC-REMBO)

To mitigate the influence of over-exploration on boundaries and random embedding distortion in REMBO, inspired by [3], we improve on [22] by combining REMBO with a new local-inspection method. The idea is that a separate optimizer inspects new optimum (e.g., new minimizer with lower value among existing samples) around the local optimum when running REMBO (main optimizer). If the inspection finds a new minimizer or reaches a maximum number of inspection iterations, then stop the inspection, and the inspection result, e.g., new minimizer is used to update the REMBO to sample next point. This has two effects. First, instead of lowering efficiency by asking and evaluating favored samples on boundaries, the local inspection focuses more on local optimum/promissing region that predefined sampling strategy may not cover, assuming most local optima are not on boundaries. If the total number of iterations for both inspection and REMBO is fixed, it would be more efficient and/or effective by increasing exploration near local optimum and reducing over-exploration near boundaries. Second, it mitigates the random embedding distortion issue of REMBO. Consequently, it helps escape from local optima in nonconvex optimization problem in electronic design [6, 14]) and provides a family of feasible solutions near global optimum. However, the inspection

Algorithm 1 IC-REMBO **Input**: Embedding_sizes, R_local, N, boundaries, noise Output: model **for** t = 1, 2, ..., N **do** Sample new point y_{t+1} using acquisition function Query the feedback = $f(random matrix A \times y_{t+1})$ integrated with simulator Augment $D_{t+1} = \{D_t, (y_{t+1}, feedback)\}$ Update the main kernel (Kl) hyper-parameters **while** new local_opt & < 5 times **do** > start inspection new inspect radius = R local + scale**for** i = 1, 2, ..., 15 **do** Sample new s_{i+1} using acquisition function Query the feedback $\text{Augment } D_{i+1}^{'} = \{D_{i}^{'}, (s_{i+1}, feedback)\}$ Update the inspection kernel (Kl')Augment $D = \{D_{t+1}, D'_{i+1}\}$, and update Klend for end while end for

takes part of total search cost no matter grid search or random search. How to reduce the inspection cost? To escape from a local optimum, the inspection radius (R) on each dimensions in the search space needs to exceed the threshold calculated from objective function decomposition[3]. When *R* is small, the inspection is cheap but less likely to find the global optimum; close to the global optimum, however it is not easy to escape from local optima so a larger R is required. [3] uses a block-wise method to separate search into small blocks to avoid high inspection complexity, however, this is inefficient and/or ineffective for large intrinsic dimensions. For example, if the inspection optimizer is BO using GP model, the complexity of calculating covariance matrix inversion to update acquisition function would be $O(n^3)$, where n is the dimension size. If we reduce dimensions in inspection, the inspection complexity would be exponentially reduced. Thus, we may use a second REMBO within the inspection to reduce the inspection dimension and complexity. (Alg. 1). Thus, we entitle the proposed method Inspection-based Combo REMBO (IC-REMBO).

Fig. 3 shows how the IC-REMBO works using a 2-D example problem: In main REMBO optimization (REMBO 1), the red line represents the low dimension domain, $\bf y$ where implements the REMBO optimization over; the blue line is the projection of 1-D embedded in 2-D dimension through multiplying the acquired point in low dimension with a random matrix A, i.e., $random\ matrix(2\times1) \cdot sample\ (y^1) = {R_1 \choose R2}$, where (R1, R2) is the point in true 2-D space; IC-REMBO samples point in one dimension, y^1 then multiplies it with a predefined random matrix. After converting low dimension point, y^1 to (R1,R2), the point is rounded within boundaries and sent to the simulator (e.g., Cadence Spectre simulation platform) to get the feedback for prior model update and sequential points acquisition. As long as the REMBO 1 finds a local optimum, the REMBO 2 starts to inspect local minimizer within a defined inspection radius (a smaller $\bf y$) and update the kernels in both REMBO 1

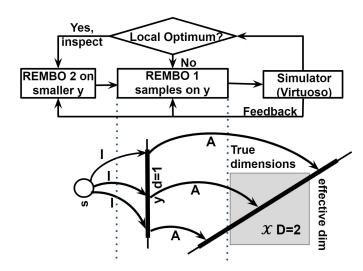


Figure 3: The Proposed Method in Circuits Design: Inspection-based Combo REMBO (IC-REMBO). Note: The minimizer inspection space, the smaller y (s space) is a 'subspace' of y space. Smaller y (s space) is defined by: 1. A smaller boundary on each dimension, 2. Reduced dimensions using embedding (e.g., true dimension χ size is 38; optimizer is sampling on a 10-dimension space, y; the inspection sampling space, s size is 4. I and A are the random matrices: $A \cdot y \in \chi$, $min(local \pm radius, A \cdot (I \cdot s)) \in \chi$). The vector in χ is send to objective function/simulator to get feedback. The samples in inspection optimizer (REMBO 2) can be used to update main optimizer (REMBO 1)'s surrogate model.

and REMBO 2. The true kernel is approximated by the REMBO 1 kernel kl.

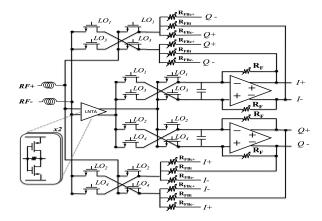
3 EXPERIMENT RESULTS

We demonstrate the effectiveness and efficiency of the proposed algorithm by comparing to state-of-the-art REMBO when optimizing a direct conversion mm-Wave receiver[4]. Fig. 4 shows the the schematic of receiver core with 38 calibration parameters (such as transistor sizes, inductor and capacitor values, feedback resistance, bias and supply voltages, etc.), 32 components, and 2 constraints. The 4 goals are high bandwidth, high conversion gain, high linearity (IIP3), and low noise figure. In the REMBO, the GP model and EI acquisition function are used empirically. We set the inspection radius at local optimum in an adaptive way. If the inspection escapes from local optimum, the R increases from initial value until reaching a pre-defined maximum number of inspection iterations, e.g., 5 times. If the inspection cannot find new optimum, then stop the inspection to save time. The optimization is integrated with the Cadence Spectre Simulation Platform.

To demonstrate the effectiveness of this approach, Table 1 compares optimization results guided by a human expert, REMBO, and the proposed IC-REMBO, all at 30 GHz. The last two rows in Table 1 are results for the proposed method after 20 and 27 iterations of

Gain IIP3 Total Main Inspection Bandwidth Noise Figure NO. Iterations Embeded Embeded (MHz) (dB) (dBm) (dB) (weighted) Simulation (Human) N/A 650 22.77 -6.9 5.57 0.195 N/A N/A **REMBO** -13.34 3.70 38->38 N/A 970 27.50 0.242 5 1 REMBO + Inspection 38->38 38->4 775 5.38 0.248 26.24 -6.87 76 2 REMBO + Inspection (Proposed) 38->10 10->4 695 23.52 -1.72 6.07 0.264 20 2 REMBO + Inspection (Proposed) 38->10 10 -> 4750 23.83 0.71 0.304 27 6.28

Table 1: The Comparisons of Optimization Results of a mm-Wave Receiver with 38 Parameters



ICREMBO optimization results

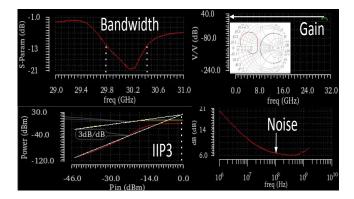


Figure 5: The Simulation Results with Optimized 38 Parameters.

optimization. IC-REMBO provides the best solution, a weighted result of 4 goals, 0.304. Compared with pre-tapeout simulation results from a human designer, the bandwidth increased from 650 MHz to 750 MHz (15.38%), and the gain increased from 22.77 dB to 23.83 dB. Note that IIP3 (with RF tones at 25 MHz and 35 MHz offset from local oscillator) can be improved from -6.9 dBm to 0.71 dBm with only a 0.5 dB increase noise figure (Fig. 5). The power consumption is 82.5 mW, also lower than that of human designer, 90 mW. The efficiency of the proposed IC-REMBO is compared with REMBO by optimizing the same receiver design. We find that REMBO needs

to try many different embeddings and struggles to obtain a satisfactory solution until we try a special case of embedding from 38 dimensions to 38 dimensions. Though this special case generates a relatively good result after trying several different embeddings, the weighted resultis low. The efficiency of sampling and possibility of covering promising region is negative influenced by the boundary over-exploration and embedding distortion issues. The proposed IC-REMBO does not need to try as many embeddings and can get a family of feasible solution after 27 optimization iterations.

Figure 6 demonstrates the effectiveness of inspection in IC-REMBO by showing one try of optimizing the mmWave receiver. Before inspection, the values of bandwidth remains at 0 MHz. As inspection starts, it starts to generate feasible solutions. The bandwidth increases gradually from 0 MHz at beginning of the optimization to/above human optimization result, 650 MHz with balanced other 3 objective goals (using weighting method). Assuming that circuit performance nearby the optimum does not vary widely, like human expert, IC-REMBO intelligently tunes selected dimensions slightly during inspection. After the inspection is finished, the optimization jumps out of local optimum (that is, it finds a new optimum) and provides a number of feasible solutions that better than human expert simulation results (see Table 1). In this way the tool quickly finds a point approximating the global optimum point (close to expert simulation results) by inspecting promising regions.

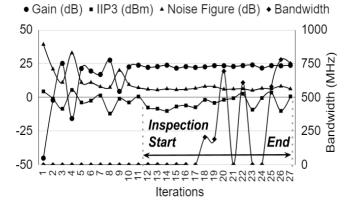


Figure 6: The Optimization Results vs Iterations

4 CONCLUSIONS

Bayesian optimization is a powerful method in solving many ubiquitous problems. However, it suffers from scaling to high dimensions. Expanding Bayesian optimization to high dimensions is of great practical interest. In this paper, the IC-REMBO algorithm is proposed. IC-REMBO improves the high dimensional optimization method, REMBO by using a new local-inspection method. To the best of our knowledge, it is the first time to apply REMBO or local inspection to electronic design. The effectiveness and efficiency are demonstrated by comparing with the state-of-the-art optimization approach when optimizing a mm-Wave receiver with 38 calibration dimensions to meet 4 goals successfully.

ACKNOWLEDGMENTS

This materials is based upon work supported by the NSF under Grant No. CNS 16-244770 - Center for Advanced Electronics through Machine Learning (CAEML) and its industry members.

REFERENCES

- James Bergstra, Dan Yamins, David D Cox, et al. 2013. Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms. In Proceedings of the 12th Python in science conference, Vol. 13. Citeseer, 20.
- [2] Eric Brochu, Tyson Brochu, and Nando De Freitas. 2010. A Bayesian interactive optimization approach to procedural animation design. In Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation. 103–112.
- [3] Yifan Chen, Yuejiao Sun, and Wotao Yin. 2019. Run-and-Inspect Method for nonconvex optimization and global optimality bounds for R-local minimizers. *Mathematical Programming* 176, 1 (2019), 39–67.
- [4] Jacob Dean, Sandeep Hari, Avinash Bhat, and Brian A Floyd. 2021. A 4-31GHz Direct-Conversion Receiver Employing Frequency-Translated Feedback. In ESS-CIRC 2021-IEEE 47th European Solid State Circuits Conference (ESSCIRC). IEEE, 187–190.
- [5] Josip Djolonga, Andreas Krause, and Volkan Cevher. 2013. High-dimensional gaussian process bandits. In Neural Information Processing Systems.
- [6] Fatemeh Kashfi, Safar Hatami, and Massoud Pedram. 2011. Multi-objective optimization techniques for VLSI circuits. In 2011 12th International Symposium on Quality Electronic Design. IEEE, 1–8.
- [7] Cheng Li, Sunil Gupta, Santu Rana, Vu Nguyen, Svetha Venkatesh, and Alistair Shilton. 2018. High dimensional bayesian optimization using dropout. arXiv preprint arXiv:1802.05400 (2018).
- [8] Chun-Liang Li, Kirthevasan Kandasamy, Barnabás Póczos, and Jeff Schneider. 2016. High dimensional Bayesian optimization via restricted projection pursuit models. In Artificial Intelligence and Statistics. PMLR, 884–892.
- [9] Wenlong Lyu, Fan Yang, Changhao Yan, Dian Zhou, and Xuan Zeng. 2018. Batch bayesian optimization via multi-objective acquisition ensemble for automated analog circuit design. In *International conference on machine learning*. PMLR, 3306–3314.
- [10] Mohit Malu, Gautam Dasarathy, and Andreas Spanias. 2021. Bayesian Optimization in High-Dimensional Spaces: A Brief Survey. In 2021 12th International Conference on Information, Intelligence, Systems & Applications (IISA). IEEE, 1–8.
- [11] Riccardo Moriconi, Marc Peter Deisenroth, and KS Sesh Kumar. 2020. High-dimensional Bayesian optimization using low-dimensional feature spaces. Machine Learning 109, 9 (2020), 1925–1943.
- [12] Riccardo Moriconi, KS Sesh Kumar, and Marc Peter Deisenroth. 2019. Highdimensional bayesian optimization with manifold gaussian processes. (2019).
- [13] Amin Nayebi, Alexander Munteanu, and Matthias Poloczek. 2019. A framework for Bayesian optimization in embedded subspaces. In *International Conference on Machine Learning*. PMLR. 4752–4761.
- [14] Hong Qian, Yi-Qi Hu, and Yang Yu. 2016. Derivative-Free Optimization of High-Dimensional Non-Convex Functions by Sequential Random Embeddings.. In IJCAI. 1946–1952.
- [15] Joaquin Quinonero-Candela and Carl Edward Rasmussen. 2005. A unifying view of sparse approximate Gaussian process regression. The Journal of Machine Learning Research 6 (2005), 1939–1959.
- [16] Philipp-Immanuel Schneider, Xavier Garcia Santiago, Carsten Rockstuhl, and Sven Burger. 2017. Global optimization of complex optical structures using Bayesian optimization based on Gaussian processes. In *Digital Optical Technolo*gies 2017, Vol. 10335. International Society for Optics and Photonics, 103350O.
- [17] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. 2012. Practical bayesian optimization of machine learning algorithms. Advances in neural information processing systems 25 (2012).
- [18] Hakki Mert Torun and Madhavan Swaminathan. 2019. High-dimensional global optimization method for high-frequency electronic design. IEEE Transactions on Microwave Theory and Techniques 67, 6 (2019), 2128–2142.
- [19] Hakki Mert Torun, Madhavan Swaminathan, Anto Kavungal Davis, and Mohamed Lamine Faycal Bellaredj. 2018. A global Bayesian optimization algorithm and its application to integrated system design. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 26, 4 (2018), 792–802.
- [20] Konstantinos Touloupas, Nikos Chouridis, and Paul P Sotiriadis. 2021. Local Bayesian Optimization For Analog Circuit Sizing. In 2021 58th ACM/IEEE Design Automation Conference (DAC). IEEE, 1237–1242.
- [21] Yi Wang, Paul D Franzon, David Smart, and Brian Swahn. 2020. Multi-Fidelity Surrogate-Based Optimization for Electromagnetic Simulation Acceleration. ACM Transactions on Design Automation of Electronic Systems (TODAES) 25, 5 (2020), 1–21.
- [22] Ziyu Wang, Masrour Zoghi, Frank Hutter, David Matheson, and Nando De Freitas. 2013. Bayesian optimization in high dimensions via random embeddings. In Twenty-Third international joint conference on artificial intelligence.
- [23] Miao Zhang, Huiqi Li, and Steven Su. 2019. High dimensional bayesian optimization via supervised dimension reduction. arXiv preprint arXiv:1907.08953 (2019).