An Efficient Kriging-based Constrained Multi-objective Evolutionary Algorithm for Analog Circuit Synthesis via Self-adaptive Incremental Learning

Sen Yin, Wenfei Hu, Wenyuan Zhang, Ruitao Wang, Jian Zhang and Yan Wang

School of Integrated Circuits, Tsinghua University, China

Bio

- Sen Yin
- Tsinghua University

2014.9-2018.6 received the B.S. degree in Wuhan University, Wuhan, China

2018.9-present pursuing the Ph.D. degree at the Institute of Microelectronics, Tsinghua University, Beijing, China.

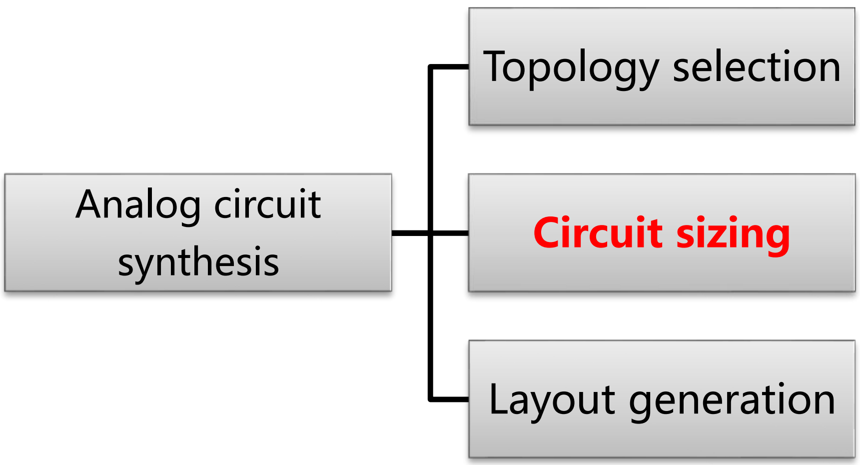
My current research interests include analog/RF circuit automatic synthesis, RF passive component modeling and the optimization of system-level circuits .

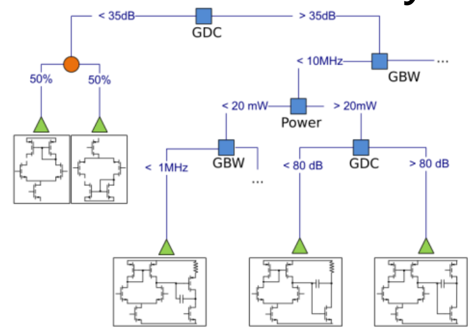
Contents

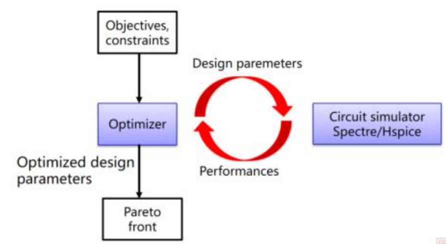
Contents

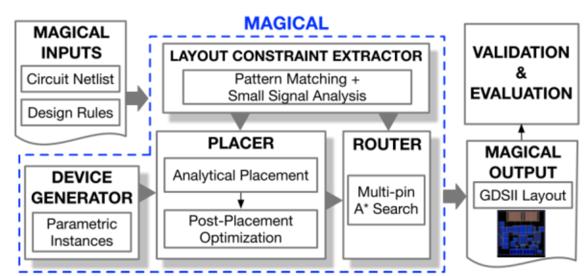
Introduction

- Manual analog circuit design is the mainstream in industry





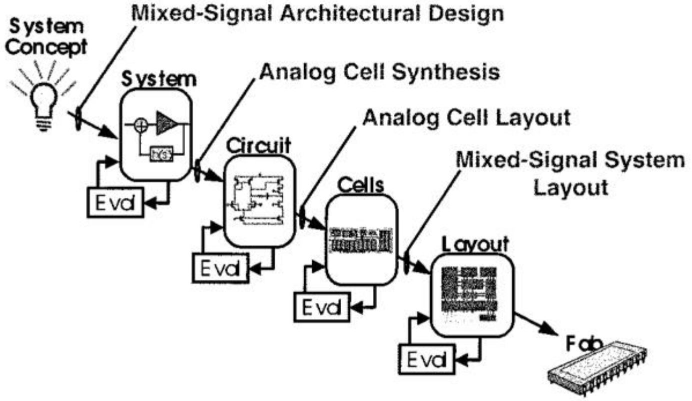




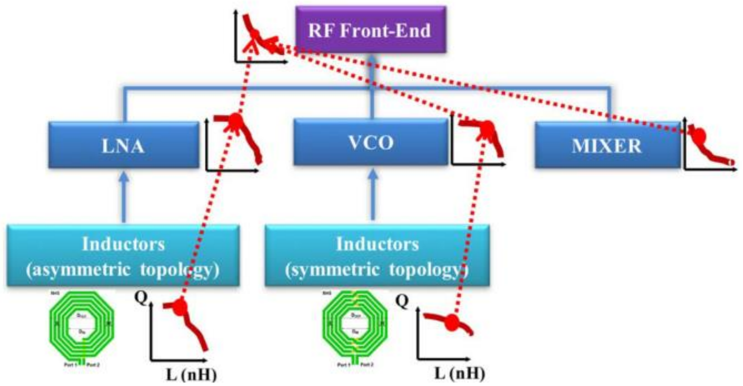
- The key problem in analog circuit synthesis: circuit sizing
- We concentrate on **analog circuit sizing**, **multi-objective optimization**.

Introduction

- Most of the work concentrates on the synthesis of analog circuits at block-level. However, few results are reported on the optimization of an analog system.
- The optimization of an analog system



Top-down method[1]



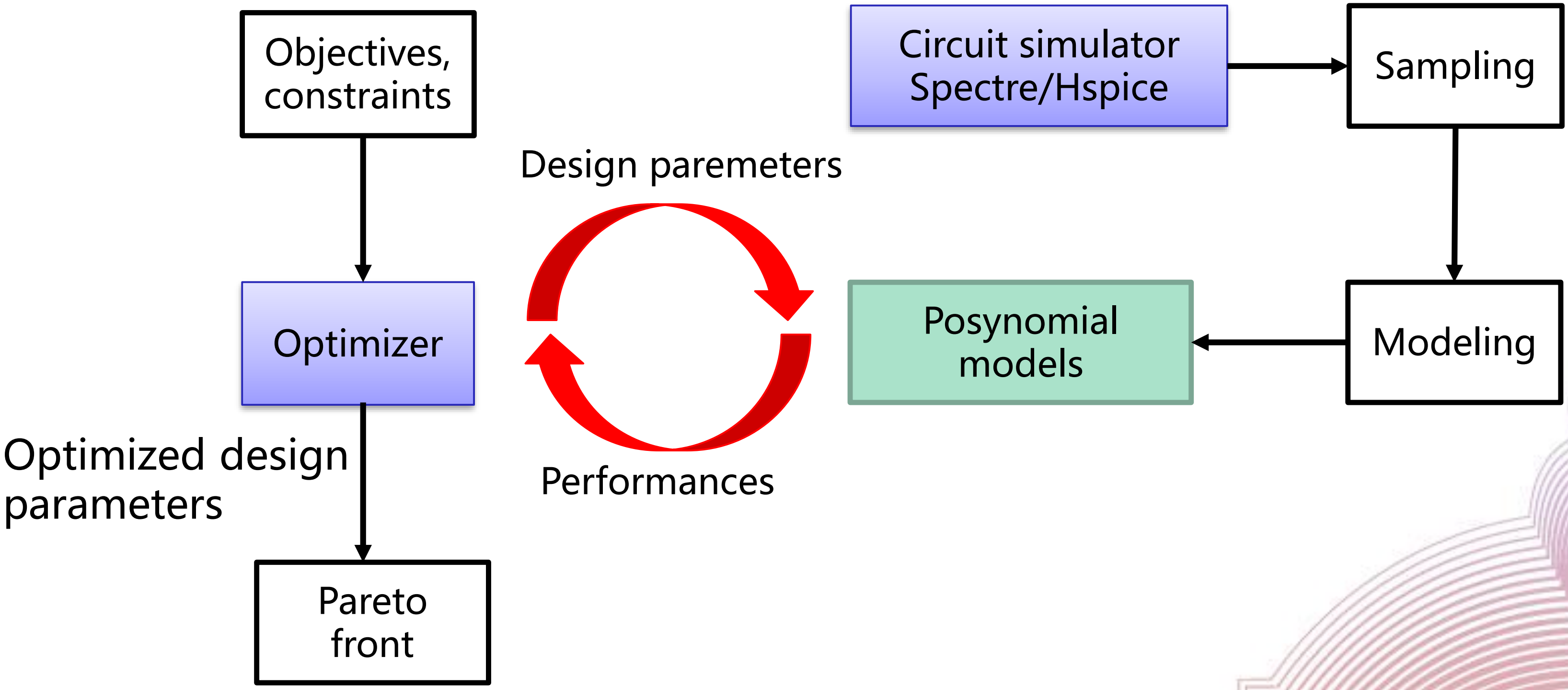
Bottom-up method[2]

- **Low efficiency** of **multi-objective optimization** is the bottleneck to optimize the **system**.

[1] Georges GE Gielen. Modeling and analysis techniques for system-level architectural design of telecom front-ends.IEEE Transactions on Microwave Theory and Techniques, 50(1):360–368, 2002.
[2] Fábio Passos, Elisenda Roca, Javier Sieiro, Rafaella Fiorelli, Rafael Castro-López, José Mar ıa López-Villegas, and Francisco V Fernández. A multilevel bottom-up optimization methodology for the automated synthesis of RF systems.IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 39(3):560–571, 2019.

Introduction

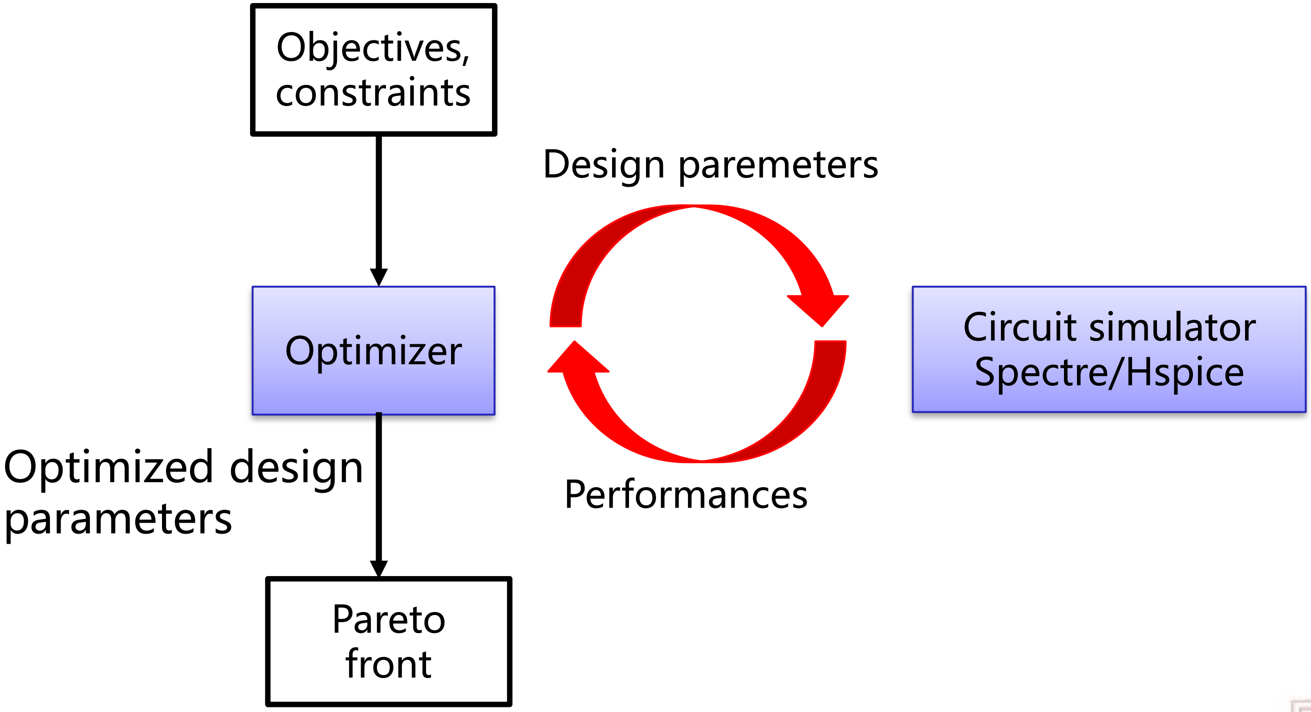
- Multi-objective analog circuit sizing
- 1.model-based methods，2.simulation-based methods



- Fast, but inaccuracy.

Introduction

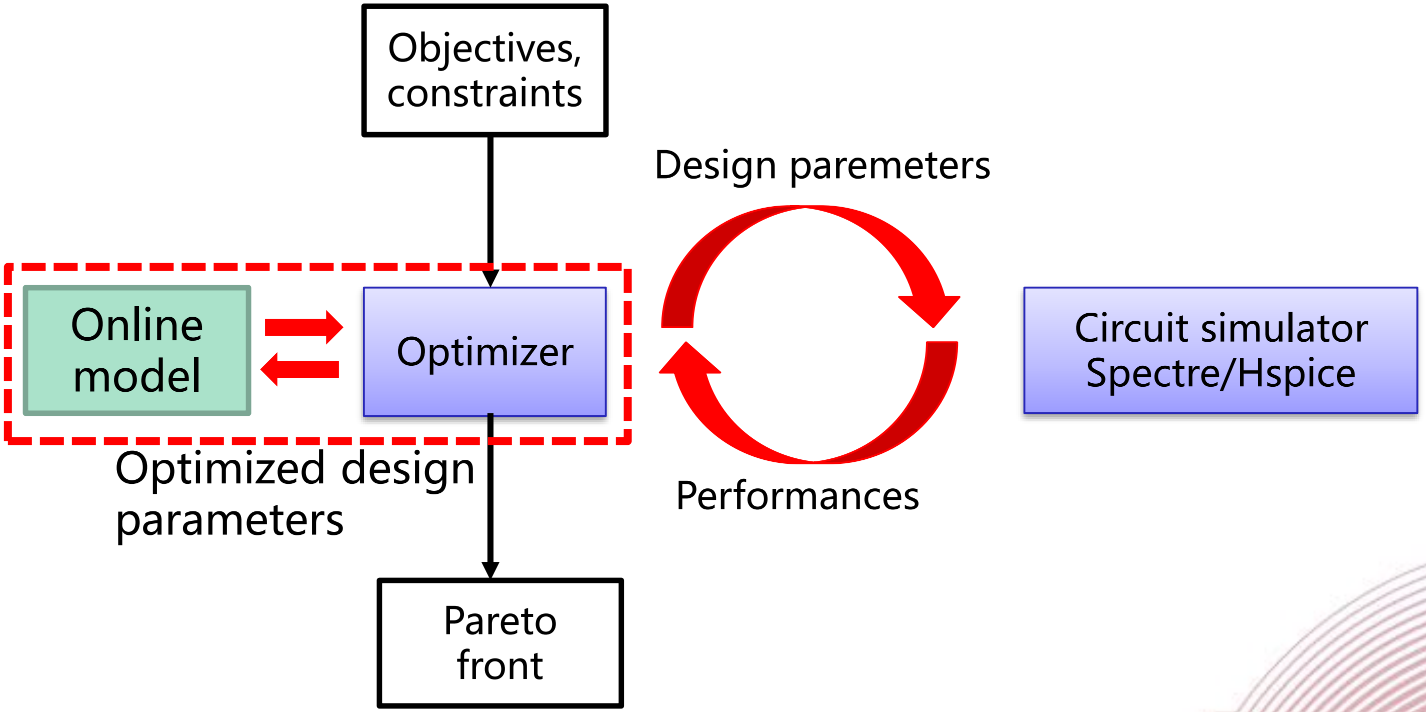
- Multi-objective analog circuit sizing
- 1.model-based methods，2.simulation-based methods



- High accuracy. Require a large number of time-consuming simulations.

Introduction

- Multi-objective analog circuit sizing
- Online model-based methods



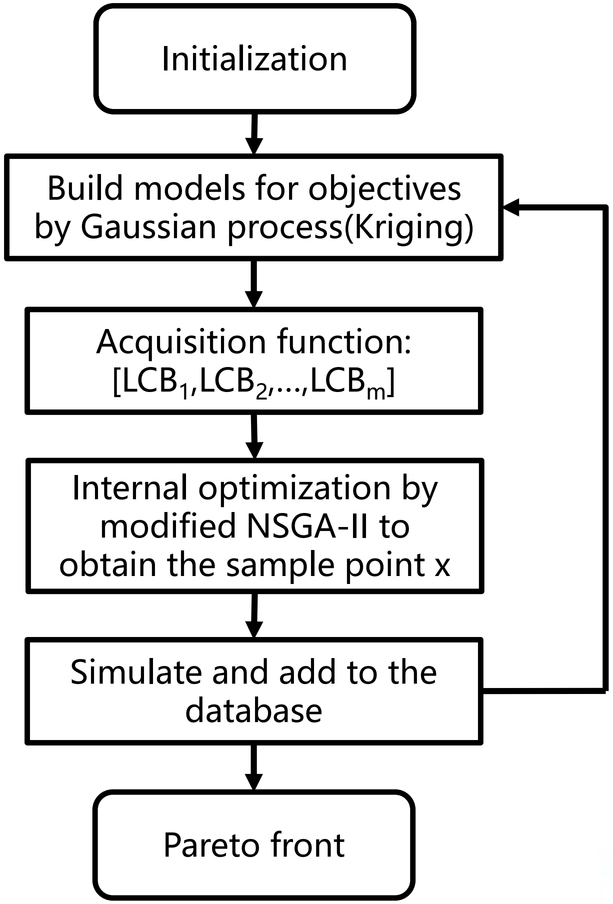
- High accuracy with less number of simulations

Contents

- 1. Introduction
- 2. Motivation
- 3. Proposed SILE algorithm
- 4. Experimental results
- 5. Conclusion

Motivation

- Multi-objective Bayesian optimization (MOBO)[3] is one of the most representative algorithms in online model-based methods

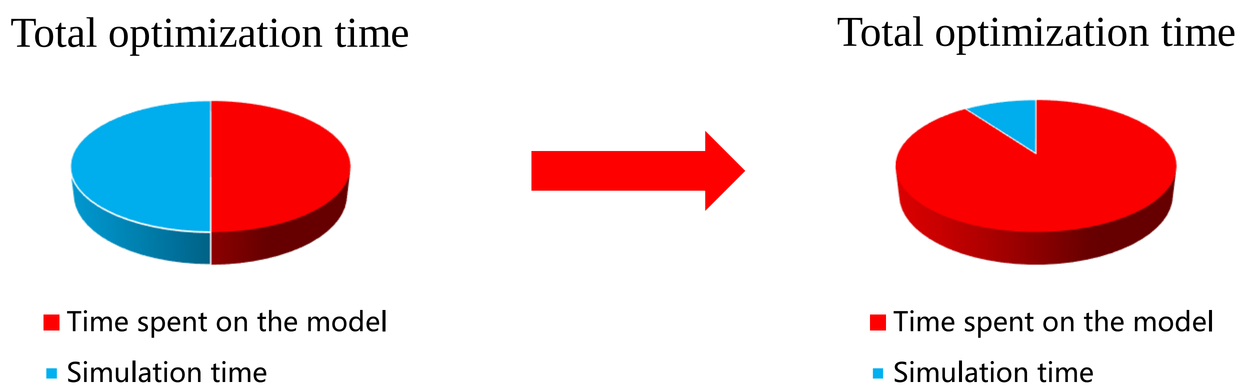


[3] Wenlong Lyu, Fan Yang, Changhao Yan, Dian Zhou, and Xuan Zeng Multi-objective Bayesian optimization for analog/RF circuit synthesis. InProceedings of the 55th Annual Design Automation Conference, pages 1–6, 2018.

- Sacrifice **the time spent on the model** for simulation time.

Motivation

- There are two problems remains to be solved in online model-based methods.
- 1. The time spent on the model is comparable to or even exceed the simulation time in online model-based methods.
- 2. Most online model-based methods can not deal with constrained problems.



- The motivation of this work is to develop an online model-based method which shortens the time spent on the model.

Contents

Proposed SILE algorithm

- Time complexity of Kriging model
- Training process: the goal is to find the fittest hyperparameters $\hat{\theta}$ for observed points. The time complexity is $\boldsymbol{O(T_1 \cdot n^3)}$. T_1 is the number of evaluating (1)

$$\hat{\theta} = \operatorname{argmax}\left(-\frac{n}{2} ln\hat{\sigma}^2 - \ln|\boldsymbol{R}|\right)$$

$$\begin{cases} \hat{\mu} = \dfrac{\mathbf{1}^T \boldsymbol{R}^{-1} y}{\mathbf{1}^T \boldsymbol{R}^{-1} \mathbf{1}} \\ \hat{\sigma} = \dfrac{(\boldsymbol{y} - \mathbf{1}\hat{\mu})\boldsymbol{R}^{-1}(\boldsymbol{y} - \mathbf{1}\hat{\mu})}{n} \\ R(x, x') = \exp\left(-\Sigma_{i=1}^{d} \theta_i (x_i - x_i')^2\right) \end{cases} \quad (1)$$

- Prediction process: Given estimated $\hat{\theta}$ and calculated R^{-1} , one can predict the performances at any untested point. The time complexity is $\boldsymbol{O(T_2 \cdot n^2)}$.

$$\begin{cases} \hat{y}(\boldsymbol{x}) = \hat{\mu} + \boldsymbol{r}^T \boldsymbol{R}^{-1} (\boldsymbol{y} - \mathbf{1}\hat{\mu}) \\ \hat{s}^2(\boldsymbol{x}) = \hat{\sigma}^2 [1 - \boldsymbol{r}^T \boldsymbol{R}^{-1} \boldsymbol{r} + \dfrac{(1 - \mathbf{1}^T \boldsymbol{R}^{-1} \boldsymbol{r})^2}{\mathbf{1}^T \boldsymbol{R}^{-1} \mathbf{1}}] \end{cases} \quad (2)$$

Proposed SILE algorithm

- We propose an efficient Kriging-based constrained multi-objective evolutionary algorithm for analog circuit synthesis via self-adaptive incremental learning (**SILE**).

Total optimization time

- Simulation time — The prescreening strategy
- Training time of the model — $O(T_1 \cdot n^3) \xrightarrow{\text{Self-adaptive strategy}} O(1 \cdot n^3) \xrightarrow{\text{The Incremental learning technique}} O(n^2)$
- Prediction time of the model — $O(T_2 \cdot n^2) \xrightarrow{\text{No internal optimization}} O(1 \cdot n^2)$

Proposed SILE algorithm

- Incremental learning technique
- How to calculate new $\tilde{R}^{-1}$ from R^{-1} of the old model?

$$\tilde{R} = \left[\begin{array}{cccc|c} R_{11} & R_{12} & \cdots & R_{1n} & R_{1(n+1)} \\ R_{21} & R_{22} & \cdots & R_{2n} & R_{2(n+1)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ R_{n1} & R_{n2} & \cdots & R_{nn} & R_{n(n+1)} \\ \hline R_{(n+1)1} & R_{(n+1)2} & \cdots & R_{(n+1)n} & R_{(n+1)(n+1)} \end{array}\right]$$

$$= \left[\begin{array}{c|c} R & A \\ \hline A^T & B \end{array}\right]$$

$$\tilde{R}^{-1} = \begin{bmatrix} R^{-1} + \boxed{R^{-1}} A C^{-1} A^T R^{-1} & -R^{-1} A C^{-1} \\ -C^{-1} A^T R^{-1} & \boxed{C^{-1}} \end{bmatrix}$$

$$C = B - A^T R^{-1} A$$

- C is a 1×1 matrix. The time complexity of computing C^{-1} is $O(1)$.
- Now that R^{-1} is known, the time complexity of $\tilde{R}^{-1}$ is $O(n^3) \rightarrow \boldsymbol{O(n^2)}$

Proposed SILE algorithm

- $\tilde{R}$ is a symmetry positive definite matrix $\leftrightarrow \tilde{R} = \tilde{L}\tilde{L}^T$
- How to calculate new $\tilde{L}$ from L of the old model?

$$\tilde{R} = \tilde{L}\tilde{L}^T \quad + \quad \tilde{L} = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix}$$

$$\Downarrow$$

$$\tilde{R} = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} L_{11}^T & L_{21}^T \\ 0 & L_{22}^T \end{bmatrix} = \begin{bmatrix} L_{11}L_{11}^T & L_{11}L_{21}^T \\ L_{21}L_{11}^T & L_{21}L_{21}^T + L_{22}L_{22}^T \end{bmatrix}$$

$$\Updownarrow$$

$$\tilde{R} = \begin{bmatrix} R & A \\ A^T & B \end{bmatrix}$$

$$\nearrow$$

$$\begin{cases} L_{11}L_{11}^T = R \\ L_{11}L_{21} = A \\ L_{21}L_{21}^T + L_{22}L_{22}^T = B \end{cases}$$

$$\Downarrow$$

$$\tilde{L} = \begin{bmatrix} \boxed{L} & 0 \\ A^T L^{-T} & \boxed{Chol(B - A^T L^{-T} \boxed{L^{-1}A})} \end{bmatrix}$$

- the time complexity of $L^{-1}A$ is $O(n^2)$ by using back substitution method
- The time complexity of $Chol(B - A^T L^{-T} L^{-1} A)$ is $O(1)$.
- Now that L is known, the time complexity of $\tilde{L}^{-1}$ is $O(n^3)$ $\rightarrow \boldsymbol{O(n^2)}$

Proposed SILE algorithm

- Self-adaptive strategy
- 1. In most cases, we build models with incremental Kriging model without updating hyperparameters. We only learn hyperparameters with Kriging model under a specific number of simulations.
- 2. In the early stage of optimization, we need to update hyperparameters more frequently.
- 3. In the later stage of optimization, we lower the frequency to update hyperparameters.

Proposed SILE algorithm

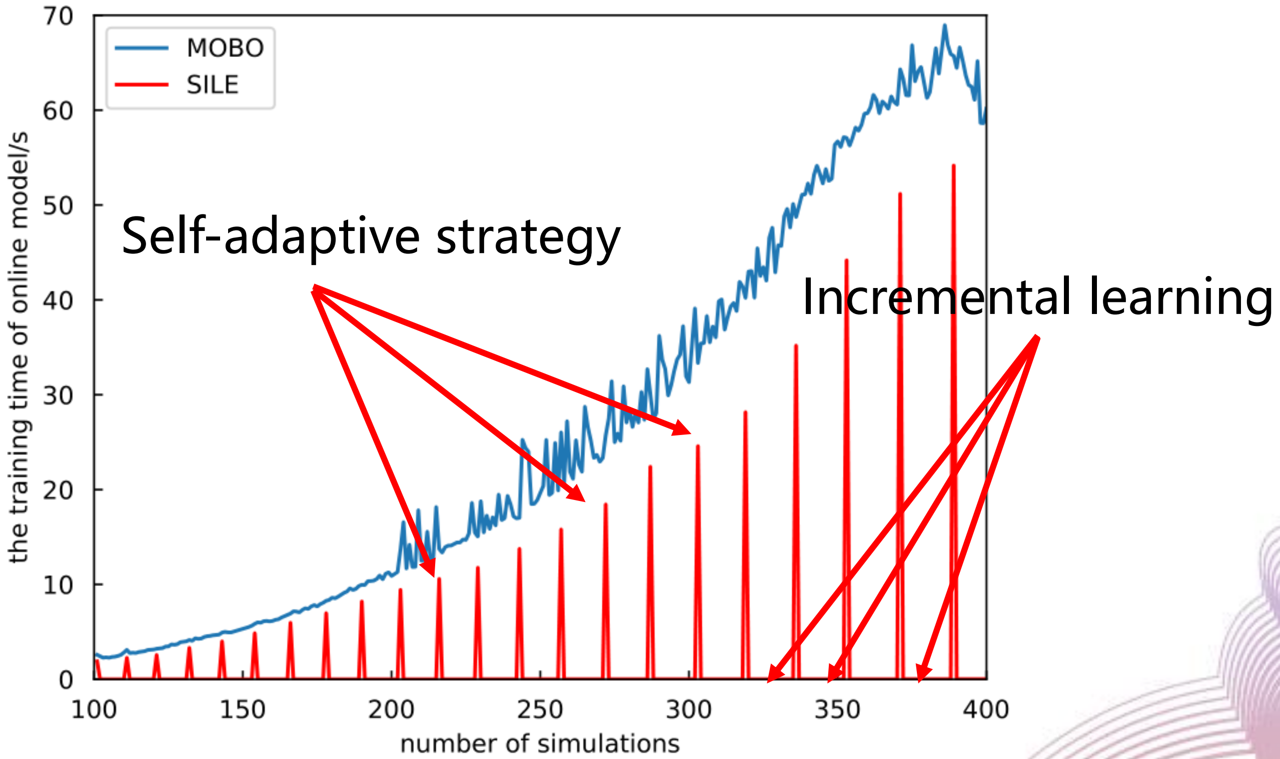
- Self-adaptive strategy
- As a result, hyperparameters are updated every H generations. H is adaptively adjusted based on the number of simulations, N

$$H = \left\lfloor \frac{N-\lambda}{N_{\max}-\lambda}(H_{\max}-H_{\min}) \right\rfloor + H_{\min}$$

- λ: the initial sample number $\quad$ $N_{\max}$:the maximum number of simulations
- $H_{\max}$: upper bound for H $\quad$ $H_{\min}$: lower bound for H
- For $\lambda = 100, N_{\max} = 200, H_{\min} = 10, H_{\max} = 19$, hyperparameters only update when $N = 100, 110, 121, 133, 146, 160, 175, 192$.

Proposed SILE algorithm

- Self-adaptive strategy



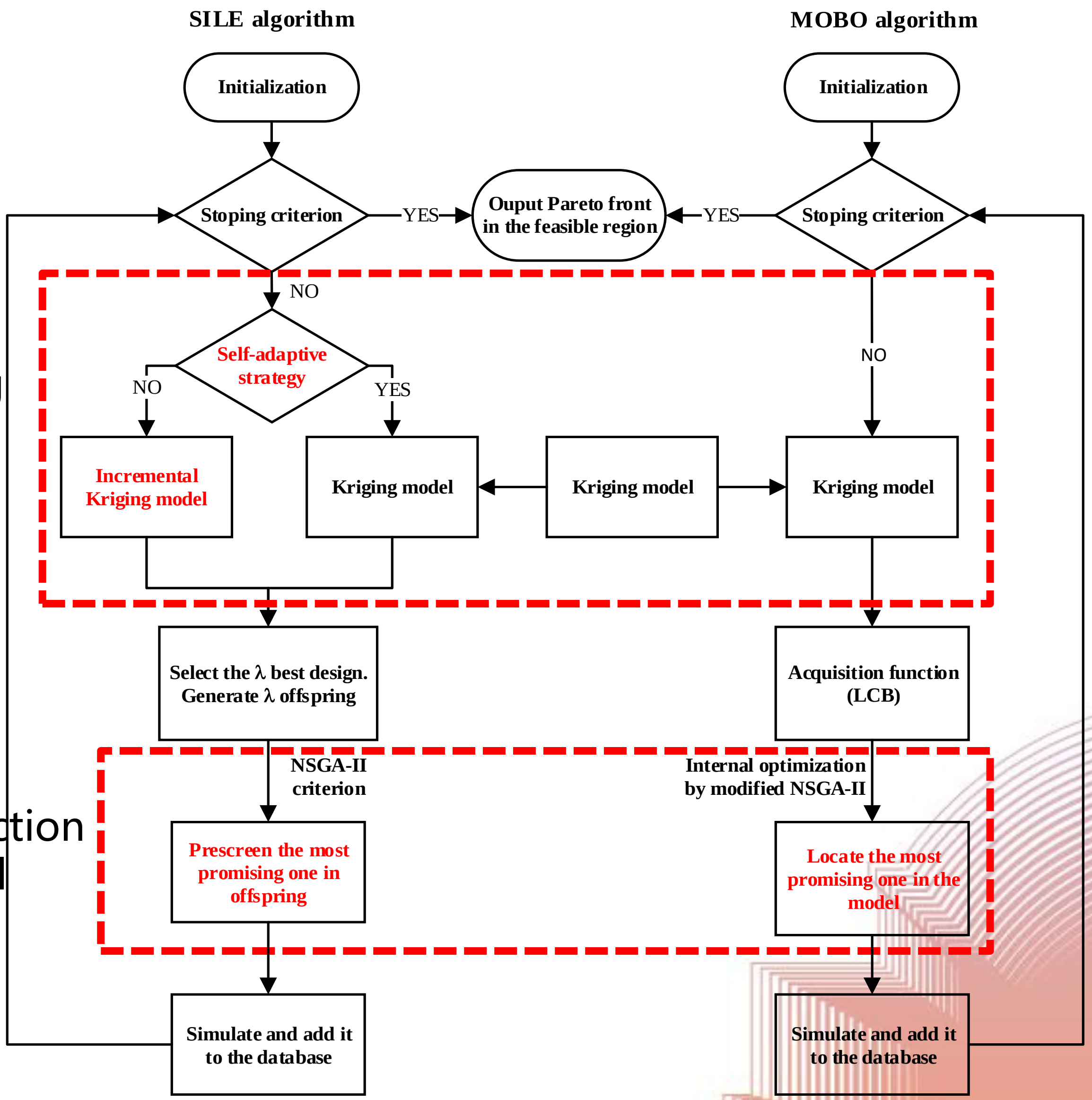
- The area under a curve is the total training time in the process of optimization
- At small spikes, we use Kriging model to optimize hyperparameters
- In the rest number of simulations, incremental Kriging model is used

Proposed SILE algorithm

The framework of prescreening

Reduce the training time of the model

Reduce the prediction time of the model



Proposed SILE algorithm

- More details
- x is said to constraint-dominate y if the following condition holds:
- 1) if $CV(x) = 0$ and $CV(y) = 0, \forall i \in \{1, 2, \ldots, m\}$ such that $f_i(x) \leq f_i(y)$ and $\exists j \in \{1, 2, \ldots, m\}$ such that $f_i(x) < f_i(y)$
- 2) otherwise, $CV(x) < CV(y)$

$$\begin{array}{ll} \text{minimize} & f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x}) \\ \textbf{s.t.} & g_i(\boldsymbol{x}) < 0 \quad \forall i \in 1{,}2, \ldots, p \end{array}$$

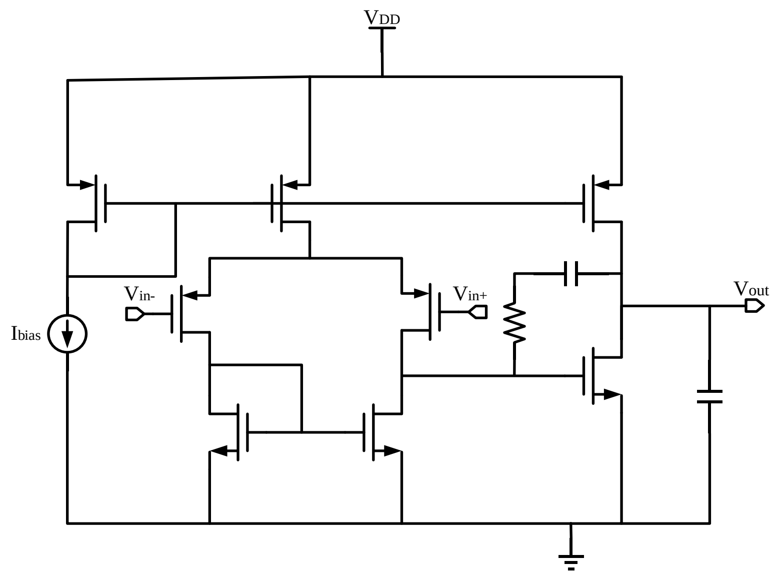
$$CV(x) = \sum_{i=1}^{p} \max(g_i(x), 0)$$

Contents

Experimental results

- A two-stage amplifier, 11 design variables, 180nm
- 15 corners: $-40°C, 27°C, 85°C$ and tt, ss, ff, fs, sf

$$minimize(-Gain, -UGBW, -PM)$$
$$s.t.\ PM > 60°$$



Experimental results

- A two-stage amplifier, 11 design variables, 180nm

Algorithm	SILE	MOBO	NSGA-II	MOEA/D
Max Gain(dB)	**81.58**	81.33	80.31	81.17
Max UGBW(MHz)	**19.64**	18.86	16.95	17.68
Max PM(°)	**93.10**	92.71	92.84	85.99
Mean HV	**14821**	13951	13880	13709
Median HV	**14726**	14038	14231	13582
Max HV	**16268**	14624	15209	16190
Min HV	**13484**	12526	10597	12224
N_{max}	**400**	400	4000	4000
Training time/s	**56.54**	1183.76	N/A	N/A
Prediction time/s	**2.53**	798.26	N/A	N/A
Simulation time/s	**1423.37**	1492.73	15051.81	15170.12
Total time/s	**1490.35**	3476.15	15052.09	15172.64

95%

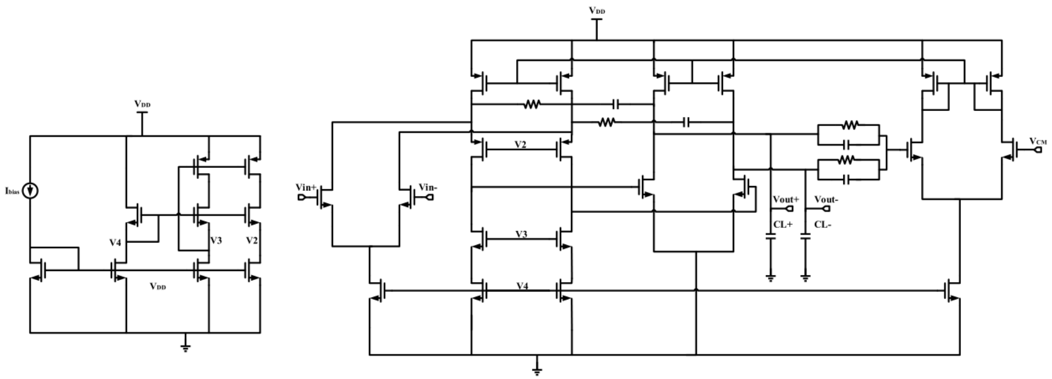
99.7%

10X

- Compared with MOBO, SILE reduce the training time by **95%** and the prediction time by **99.7%.** SILE shows a speedup of **10X** in terms of the total time while achieving better results.

Experimental results

- A fully differential operational amplifier , 21 design variables, 65nm



$$
\begin{aligned}
\text{minimize} \quad & (-Gain, -GBW) \\
\text{s.t.} \quad & PM_{dm} > 60^\circ \\
& PM_{cm} > 50^\circ \\
& GBW_{cm} > 1.2GBW_{dm} \\
& SR > 100V/\mu s \\
& overshoot_c < 15\% \\
& tset_c < 50ns \\
& overshoot_r < 15\% \\
& tset_r < 50ns
\end{aligned}
$$

Experimental results

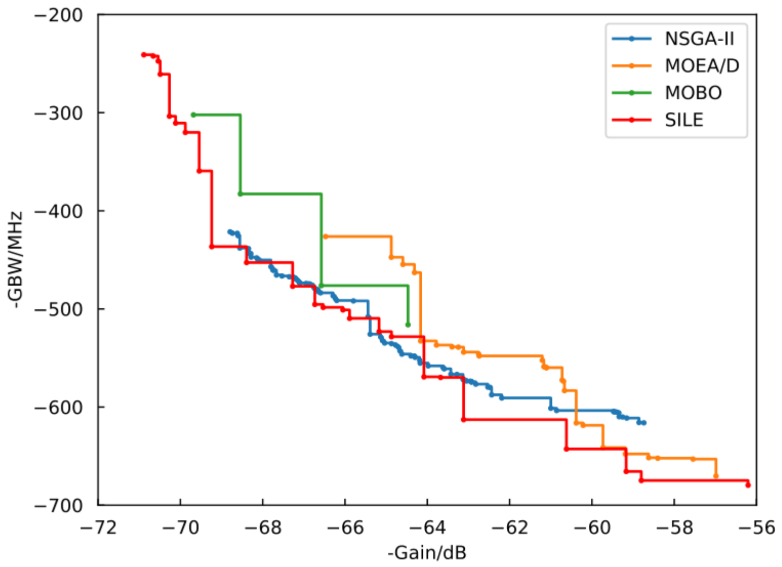
- A fully differential operational amplifier , 21 design variables, 65nm

Algorithm	**SILE**	MOBO	NSGA-II	MOEA/D
Max Gain(dB)	**70.89**	70.37	69.92	68.10
Max GBW(MHz)	**770**	639	685	738
Mean HV	**10684**	7678	9709	9368
Median HV	**10892**	7715	9565	9344
Max HV	**11676**	9235	10664	10160
Min HV	**9239**	5799	9102	8724
$N_{\max}$	**400**	400	4000	4000
Training time/s	**382.44**	8084.91	N/A	N/A
Prediction time/s	**6.29**	3014.34	N/A	N/A
Simulation time/s	**804.43**	837.67	7420.74	7234.19
Total time/s	**1200.90**	11941.38	7420.97	7235.87

95%

99.8%

6X



- SILE reduces the training time by **95%**, the prediction time by **99.8%** while achieving much better PF. There is a **6X** speedup over NSGA-II and MOEA/D regarding the total time.

Contents

Conclusion

- We propose an efficient Kriging-based constrained multi-objective evolutionary algorithm for analog circuit synthesis via self-adaptive incremental learning.

Total optimization time
- Simulation time
- Training time of the model
- Prediction time of the model

The prescreening strategy

$O(T_1 \cdot n^3) \xrightarrow{\text{Self-adaptive strategy}} O(1 \cdot n^3) \xrightarrow{\text{The Incremental learning technique}} O(n^2)$

$O(T_2 \cdot n^2) \xrightarrow{\text{No internal optimization}} O(1 \cdot n^2)$

- Experimental results on two real-world circuits demonstrate that compared with MOBO, our method can reduce the training time of Kriging model by **95%** and the prediction time by **99.7%** . Compared with NSGA-II and MOEA/D, the proposed method can achieve up to **10X** speed up.

Thanks for your attention!