Towards Machine Learning-Assisted Electronic Design Automation: Microwave Filter, Power Amplifier, and Semiconductor Device

Liyuan Xue

SUBMITTED IN FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

JAMES WATT SCHOOL OF ENGINEERING

COLLEGE OF SCIENCE AND ENGINEERING

SEPTEMBER 2024

To the intelligence,
emerged accidentally in this trivial world.

Abstract

Over the decades of development, electronic design automation (EDA) has been widely applied in most electronic design problems, especially in advanced and sophisticated digital systems. In contrast, the degree of automation for distributed-element circuits, e.g. microwave or millimeter-wave (mm-wave) devices characterized by electromagnetic (EM) simulations, and semiconductor devices characterized by technology computer-aided design (TCAD) simulations, is still very limited. Two challenges are especially notable. First, both EM and TCAD simulations are computationally expensive. Second, some design problems in these fields are highly parameter-sensitive with many local optimal solutions. Consequently, fully algorithmic EDA in these fields is still in its infancy, especially incorporating with advances of machine learning (ML) or artificial intelligence (AI) techniques for higher automation levels.

The objective of this thesis is accordingly to develop a more generic and effective framework (than hitherto) for design automation in these fields, assisted by cutting-edge progress in ML. Three representative circuits/devices are selected for investigation: microwave filter, monolithic microwave integrated circuit (MMIC) power amplifier (PA), and semiconductor devices. Beginning with a brief introduction of EDA, basic concepts of relevant optimization algorithms and ML techniques are brought in subsequently, then each topic is unfolded by a comprehensive literature review followed by details of the proposed methodology, experimental results, and comparisons. Specifically,

- **Microwave Filter:** A design automation method composed of two-phase design optimization is proposed for three-dimensional microwave filters. In each phase, the bespoke objective functions and optimization algorithm are proposed to improve the robustness and success rate. By incorporating with a programmable initial design synthesis, the proposed methodology enables the first unsupervised design automation without human intervention.
- **MMIC PA:** An efficient layout-level automated design methodology is proposed for MMIC PAs, supporting holistic characterization with EM, small- and large-signal simulations and being compatible with most foundry process design kits. Bayesian neural networks are integrated with novel hybrid local and global search strategies. Two MMIC PAs—a balanced Class-AB PA and a wideband Doherty PA—were successfully synthesized with the later taped out for manufacturing.

- **Semiconductor Device:** An attempt towards algorithmic design optimization for semiconductor devices is presented through two case studies. The first optimized the epitaxial layer of a commercial III-V pHEMT for higher cut-off and maximum oscillation frequency over terahertz, achieving a 30% and 57% improvement, respectively. The second study proposed the concept of device circuit co-optimization, enhancing the performance of a planar CMOS-based inverter to outperform several reported devices with advanced technologies.

In conclusion, this thesis investigated ML-assisted EDA within the three aforementioned areas. The research outcomes demonstrate significant improvements in design efficiency, performance, and versatility. This work paves the way for further research into higher degrees of design automation, facilitating the emergence of the upcoming AI-driven EDA era.

Contents

List of Tables

List of Figures

Acknowledgements

For a scientific researcher, writing and defending a Ph.D. thesis is perhaps the most significant event in their early career. Although in the current era, some opinions argue that much university research lags behind practical applications and may even underperform compared to company-led research—particularly in fields like artificial intelligence—my personal experience transitioning from industry back to academia leads me to believe that exploring a seminal field without presupposed constraints and writing over 100 pages of material for preserving public knowledge, rather than keeping it confidential for business interests, is undoubtedly more meaningful for the future. With this in mind, the thesis was accomplished with the help of many individuals around me, for which I am deeply grateful.

This thesis has been conducted under the primary supervision of Prof. Bo Liu, who has provided invaluable encouragement and excellent guidance throughout. Further supervision has been provided by Dr. Oluwakayode Onireti and Prof. Muhammad Ali Imran. I would like to express my sincere gratitude to the three supervisors for their generous support and help.

I would like to extend my profound gratitude to the convener Dr. Julien Le Kernec, and examiner Dr. Khiem Nguyen and Dr. Faisal Tariq for carefully examining my annual progress review.

I would like to thank my colleagues Yushi Liu and Mobayode Olusola Akinsolu for their insightful discussions on challenging issues.

I would also like to thank Jing Wang, Ankit Dixit, and Naveen Kumar for their help with TCAD simulation.

I appreciate and express my gratitude for the scholarship provided by the James Watt School of Engineering, which enabled me to undertake this Ph.D. research.

Lastly, but certainly not least, my heartfelt thanks to my parents and my girlfriend, who supported and helped me a lot when I felt frustrated, depressed, and homesick.

Declaration

Name: Liyuan Xue

Registration Number:

I certify that the thesis presented here for examination for a PhD degree of the University of Glasgow is solely my own work other than where I have clearly indicated that it is the work of others (in which case the extent of any work carried out jointly by me and any other person is clearly identified in it) and that the thesis has not been edited by a third party beyond what is permitted by the University's PGR Code of Practice.

I declare that the thesis does not include work forming part of a thesis presented successfully for another degree.

I declare that this thesis has been produced in accordance with the University of Glasgow's Code of Good Practice in Research.

I acknowledge that if any issues are raised regarding good research practice based on review of the thesis, the examination may be postponed pending the outcome of any investigation of the issues.

Liyuan Xue

Abbreviations

3D	Three-Dimensional
ACO	Ant Colony Algorithm
ADAM	Adaptive Moment Estimation
ADS	Advanced Design System
AEL	Application Extension Language
AI	Artificial Intelligence
AMAM	Amplitude Modulation to Amplitude Modulation
AMPM	Amplitude Modulation to PHASE Modulation
ANN	Artificial Neural Network
ASIC	Application-Specific Integrated Circuit
BBO	Black-Box Optimization
BFGS	Broyden–Fletcher–Goldfarb–Shanno
BNN	Bayesian Neural Network
BO	Bayesian Optimization
CAD	Computer Aided Design
CAE	Convolutional Autoencoder
CM	Coupling Matrix
CMOS	Complementary Metal-Oxide-Semiconductor Transistor
DDPG	Deep Deterministic Policy Gradient
DE	Differential Evolution
DNN	Deep Neural Network
DPA	Doherty Power Amplifier
DTCO	Design Technology Co-Optimization
EA	Evolutionary Algorithm
EDA	Electronic Design Automation
EI	Expected Improvement
ELBO	Evidence Lower Bound
EM	Electromagnetic
EO	Expensive Optimization
FBW	Fractional Bandwidth
FDTD	Finite-Different Time-Domain
FEM	Finite element method

FinFET	Fin Field Effect Transistor
FPGA	Field Programmable Gate Array
GA	Genetic Algorithm
GAA	Gate All Rround
GaAs	Gallium Arsenide
GaN	Gallium Nitride
GP	Gaussian Process
HB	Harmonic Balance
I/O	Input or Output
IC	Integrated Circuit
InP	Indium Phosphide
LCB	Lower Confidence Bound
LMBA	Load-Modulated Balanced Power Amplifier
LSTM	Long Short-Term Memory
MDP	Markov Decision Processes
MIM	Metal-Insulator-Metal
ML	Machine Learning
MLP	multilayer perceptron
mm-wave	Millimeter-Wave
MMIC	Monolithic Microwave Integrated Circuit
MoM	Method of Moments
MOSFET	Metal-Oxide -Semiconductor Field Effect Transistor
MUF	Maximum Oscillation Frequency
NM	Nelder-Mead
NN	Neural Network
NP-hard	Non-Deterministic Polynomial-Time Hard
ODE	Ordinary Differential Equation
PA	Power Amplifier
PAE	Power-Added Efficiency
PDE	Partial Differential Equation
PDK	Process Design Kit
pHEMT	Pseudomorphic High-Electron-Mobility Transistor
PI	Probability of Improvement
PSO	Particle Swarm Optimization
RF	Radio Frequency
RL	Reinforcement Learning
SAEA	Surrogate-Assisted Evolutionary Algorithm
SAO	Surrogate-Assisted Optimization
SGD	Stochastic Gradient Descent

SM	Space Mapping
SMAS	Surrogate Model-Aware Search Mechanism
SoC	System-on-Chip
SPICE	Simulation Program with Integrated Circuit Emphasis
SRFT	Simplified Real Freuency Technique
SRH	Shockley–Read–Hall
TCAD	Technology Computer-Aided Design
TR	Trust-Region
UCB	Upper Confidence Bound
VAE	Varitional Autoencoder
VLSI	Very Large Scale Integration
ZPE	Zero, Pole, and Edge

Chapter 1

Introduction

1.1 Electronic Design Automation

Electronic design automation (EDA) has a long history in the development of modern electronic industry, and it is at the center of modern technological advances in improving the quality and convenience of human lives [1]. EDA enables automated design, debugging, testing, and verification processes of electronic systems with billions of transistors to meet requirements and specifications, ensuring the Moore's Law a significant driving force rather than a mere curiosity over the past quarter-century [2]. Without EDA, many electronic devices that have changed our daily lives, such as laptops, smartphones, video games, and other equipment that we now take for granted, would be inconceivable. As a remarkable success in design technology [3] (i.e. technology focus on designing and developing products), EDA has driven and supported numerous applications across various complex and sophisticated systems and will continue to evolve alongside advances in electronic and microelectronic technology.

As the name suggests, EDA employs a set of tools, algorithms, methodologies, and infrastructure to streamline and speed up the electronic design process while reducing human effort at the same time. The most impactful and successful use of EDA is in digital very large scale integration (VLSI), wherein EDA tools are essential in designing intricate integrated circuits (ICs) from logic to layout, allowing for the implementation of tens of billion transistors [4]. The history of these automation tools can be briefly summarized. After half a century of significant research into understanding semiconductor physics and simulating circuit characteristics, many EDA tools for chip design emerged in the 1960s to replace manual and laborious work [5]. Several now

well-known companies, such as Cadence and Synopsys were founded, and many key techniques were developed and standardized (e.g., logic synthesis and timing analysis) in that era [6]. Entering the new millennium, the IC design flow underwent a series of paradigm shifts as the feature size of transistors decreased to below 100 nm. This denser integration introduces additional uncertainty due to manufacturing process variations. More analysis functions were integrated at different abstract levels to mitigate performance degradation and improve product yields. These methodologies form the essential EDA infrastructure and now enable the design of very large digital systems of logic or memory, including but not limited to system-on-chip (SoC), application-specific integrated circuits (ASICs), and field programmable gate array (FPGA).

Unlike the significant prosperity of EDA in digital ICs as mentioned above, automated design for analog or distributed-element circuits (i.e., circuits whose dimensions are comparable to or larger than the wavelengths of the transmitted signals. For instance, circuits used for radio frequency (RF), microwave, or millimeter-wave (mm-wave) applications are of such type. Referred to hereinafter as distributed circuits) and semiconductor devices is relatively limited, and not as standardized and generic as in digital circuits until the modern era. Specifically, analog ICs are still largely designed and laid out manually, with limited assistance from automation tools [2]; humans remain dominant in this process. Distributed circuits or devices (e.g., microwave filter or antenna) still heavily rely on the understanding of the analytic analysis of electromagnetic (EM) fields, which helps designers capture the *feel* (i.e. intuition) behind specific structures for designing more intricate devices [7], as well as on the experience of seasoned engineers. As for semiconductors, numerical simulation for semiconductor devices often lags behind their fabrication and experimental measurements; most new technologies are invented, designed, and validated in laboratories or foundries based on their manufacturing experience rather than with numerical simulation techniques [8].

Due to this, other terms are widespread in the literature, emphasizing more on the interaction between humans and computers with less focus on automation capability—computer-aided design (CAD) [9] and technology computer-aided design (TCAD), the latter being for semiconductors. In this thesis, however, it is argued that both EDA and CAD represent a long journey to pursue rather than merely a destination. Therefore, no distinction is made between EDA and CAD for electronic design, as their goals are the same—to liberate people from the laborious work of product design while reducing time-to-market.

Although the concept of CAD and TCAD for distributed circuits and semiconductor devices has gone through the same decades of development, there are at least three factors impeding a higher automation level in these areas, ranging from physics to practice. First, it is important to consider whether the underlying physics—the scientific principles—are fully acknowledged and understood. These fundamentals are often described by equations that are either linear or partial differential. Second, it is important to note whether effective numerical methods arbitrary structures and configurations of devices of interest are well established. This is the foundation of any accurate characterization or simulation. Lastly, it is crucial to consider whether there are key design processes that are difficult to implement by machines. This determines the feasibility of high-level design automation. To narrow our scope to the theme of this thesis, we briefly analyze the above three factors with respect to circuits and devices in the different fields.

In terms of digital ICs, it is broadly true that their basis lies in Boolean algebra. Therefore, by setting sufficient noise and power margins on MOS transistors, design activities can be conducted at a series of independent, abstract, and analytic levels. This facilitates the use of automation tools, whether rule-based (i.e., setting a set of rules for all possible scenarios) or combinatorial optimization-based (i.e., optimizing a series of discrete programming problems using algorithms), enabling a high degree of design automation.

Analog circuits complicate this situation, wherein transistors are no longer identical but have different configurations [10]. Simulation program with integrated circuit emphasis (SPICE) and SPICE-like simulators (e.g. HSPICE [11] and Spectre [12]) based on Kirchoff's law become pivotal in response analysis. The mathematics behind these simulators involves solving linear and ordinary differential equations (ODEs), the latter for transient analysis in the time domain. Although their computational cost is higher than that of digital ICs, it remains acceptable as results can often be obtained within seconds. Nevertheless, while all specifications may be met in the topology design of analog ICs, performance can be moderately degraded in its physical design (i.e. designing the layout) due to non-ideal effects of silicon materials and interconnects. Post-layout SPICE simulation, including parasitic extraction, is thus indispensable. When the resulting performance with parasitics is intolerable at this stage, redesign and iterative manual tuning are often inevitable. This iterative process relies mostly on the experience of designers, greatly hindering the implementation of a higher level of design automation.

Distributed circuits and devices are even more complicated to accurately characterize due to the need to solve partial differential equations (PDEs), specifically, Maxwell's equations regarding electromagnetic (EM) fields. The solution is analytic in closed form only in simple cases and under assumptions, necessitating specific numerical approaches for general cases [13]. Three representative full-wave numerical simulation methods, finite-different time-domain (FDTD), method of moments (MoM, referred to hereinafter as momentum method), and finite element method (FEM), were developed and matured in the 1990s [14]. FDTD and FEM are superior for three-dimensional (3D) structures such as antennas or cavity-based filters, while MoM has advantages in designing single- or multi-substrate quasi-planar circuits [15]. However, these methods pose a significant computational burden even on powerful workstations; it is common to observe that a simulation of a moderate structure can take several to tens of minutes, making this approach often a final simulation resort (i.e., conducting EM simulation) before fabrication. Although lumped equivalent circuits are proposed and can be applied as an alternative during the design process, their accuracy diminishes with increasing frequency. Therefore, exhaustive exploration of the design space using EM simulation is infeasible. Intuition in EM theory with specific circuits and structures from experienced engineers becomes the core of design processes.

In terms of semiconductors, TCAD is still in its infancy [16]. The primary performance boosters of transistors over the past decades, such as new device structure, doping strategies, and size shrinking, were proposed and verified by intensive experiments rather than comprehensive numerical simulation [8, 17]. This is primarily attributed to two reasons: the complexity of multi-physics interaction in semiconductor devices, such as the mechanical stress/strain of lattices and electron transport [16], and the significant effect of fabrication processes on these devices including annealing and implantation [18]. Thus, structures were implemented and tested on a case-by-case basis, while accurate simulation became rather intricate and costly, especially in the absence of general simulators. Until now, simulation and modeling of some semiconductor devices have remained the core topic within the device research community [19, 20, 21]. There is a lack of algorithmic design methodologies for structural exploration, not to mention the automation aspect.

Table 1.1 summarizes the main points of the above discussion. As the types of circuits range from digital to semiconductor devices, the time cost and complexity of numerical simulations escalate, while the level of automation decreases. Therefore, it is expected that research in the EDA of distributed circuits and semiconductor devices has the potential to lead to notable improvements in performance and efficiency, especially

Table 1.1: Comparison of EDA for different areas

Area	**Physics**	**Mathematics**	**Simulation time cost**	**Degree of auto.**
Digital ICs	-	Boolean algebra, Graph theory	Low	High
Analog circuits/ICs	Kirchoff's law	Linear equations	Medium low	Moderate
Distributed circuits/ICs	Maxwell's equation	Partial differential equation	High	Low
Semiconductor devices	Schrödinger's equation, Boltzmann transport equation	Partial differential equation	Very high	None

considering the involvement of machine learning techniques, which makes the reduction of numerical cost possible. Henceforth, our attention will be dedicated entirely to EDA in these areas. Two typical distributed circuits or devices, namely microwave filters and power amplifiers (PAs), are selected for further investigation. Microwave filter is a representative distributed circuit or device. PA can be considered both an analog and a distributed circuit when its operating frequency is high. For semiconductor devices, the focus is on two types of transistors used in different applications: the III/V pHEMT and the CMOS with an inverter circuit. Further discussion regarding challenges and objectives will be detailed in Section 1.3. Prior to this, the background of machine learning in EDA is introduced.

1.2 Machine Learning in Electronic Design Automation

Machine learning (ML) has become the most prominent technique in this era, which empowers and will continue to boost many cutting-edge technologies, such as automatic drive, personal assistant [22], and smart manufacturing [23], liberating people from laborious decisions and operations. As the major branch of artificial intelligence (AI), ML demonstrates incredible capabilities in classification, regression, detection, and design space exploration [24]. Since these tasks are also quite common in developing design automation techniques, much interest has been gained by both industrial and academic communities [25].

As a continuously evolving field, ML can be broadly categorized into three types by the forms of learning: supervised learning, unsupervised learning, and reinforcement learning [26]. Supervised learning, among them, is the foundational one [27]. It learns the mapping and constructs an analytic model under given input-output pairs by minimizing errors between the model output and the ground truth. Classification and regression are two principal problems in supervised learning, which are distinguished by their output data types: classification deals with categorical variables, while regression is concerned with continuous prediction. ML models trained through a supervised approach can range from relatively simple techniques, such as linear regression, to highly complex approaches, like deep neural networks (DNNs) (also known as artificial neural networks (ANNs) for some simpler structures) or kernel machines [28]. In recent years, DNNs have demonstrated their amazing capability in discovering intricate structures with large-scale datasets, powering many advances of modern research and applications including large language models, speech synthesis, and video generation. Many research efforts in EDA have also benefited from these advances.

The power of machine learning techniques has been demonstrated extensively in digital ICs [29], ranging from logic synthesis [30, 31] and physical design [32, 33], to verification [34] and testing [35]. In terms of analog circuits [36], advancements have also been achieved in ML-based circuit sizing [37, 38], layout placement [39, 40], and fault diagnosis [41]. For distributed circuits and semiconductors, research primarily focused on circuit/device modeling and ML-based device optimization, with each topic explained as follows.

The most essential and straightforward application of ML in EDA is circuit/device modeling, wherein supervised learning is primarily involved. Due to the time-consuming simulation for both distributed circuits and semiconductor devices, finding cheaper performance evaluators is often necessary. In the early period, many developments of device modeling were made by building equivalent circuit abstraction according to corresponding theories [9]. When some machine learning techniques exhibit competitive effectiveness compared to existing approaches, shifts are instantly captured by the research community [42]. For instance, ANN modeling for microwave circuits can date back to the 1990s [43, 44], where passive and active components, such as microstrip line [45, 46] and spiral inductor [47], were modeled by ANNs for high-level design. A similar trend also occurred for semiconductors but lagged until the 2020s [48, 49, 50], with the capacity of ML-based modeling being continually explored. However, in these research, ML models should often be trained case by case for specific applications. When the structure of circuits/devices becomes complex or the reusable components

are not explicit, the strength of training ML models becomes unclear. Furthermore, ML models are often criticized for their lack of interpretability. When the device technology is updated or new parameters are considered, models should be trained from scratch, which may be unaffordable in many scenarios.

While device modeling aims to accurately mimic device behavior by ML models with less numerical cost, design optimization leverages ML models as *online* surrogates, where *online* refers to updating models consecutively. Prominent achievements were obtained for microwave circuit optimization in the early 2000s [51], wherein ANNs were employed as coarse models or auxiliary models to replace computationally expensive simulations. ANNs were trained and updated during optimization, sometimes with empirical functions to accelerate training. These early methods can be unified within the framework of space mapping (SM), and they often lack explicit search engines. When local search algorithms, e.g. quasi-Newton or Trust-region, are used, SM method is hard to guarantee an optimal solution for general design problems. Recent progress involves the use of new neural network structures and the integration of transfer functions for passive components [44]. More discussion can be found in Section 3.2. The work in [52] introduced computational intelligence in designing analog and high-frequency circuits. By incorporating ML model (i.e., the Gaussian process) with global optimization algorithm, design automation is achieved for antennas and RF ICs with affordable computational cost. However, this method was only validated on the examples with simple topology; its performance for complex microwave circuits is still to be discovered. As for semiconductors, there is a lack of decisive work for device optimization.

To sum up, the development of ML in EDA is still very immature, especially for distributed circuits and semiconductors. Although some progress has been achieved on specific topics, e.g., computational intelligence methods incorporating ML techniques for circuit design optimization, there is still a significant gap towards practical problems. Much attention and development are needed to enable a higher level of design automation.

1.3 Challenges and Research Objectives

This thesis aims to explore potential pathways towards ML-assisted EDA. The research unfolds in two areas: distributed devices/circuits and semiconductor devices. For the distributed area, two representative devices and circuits are considered: microwave filter and power amplifier. Microwave filter is considered as a typical passive device characterized by EM simulation, while PA is described as the most critical active and nonlinear circuits requiring holistic characterization with multiple simulators. Note that the focus in this research is given to PAs used in RF and microwave applications, rather than those in low-frequency analog circuits. In terms of semiconductor device, transistors are recognized as the most crucial ones. Therefore, research is conducted on two typical transistors: III/V pHEMT for terahertz applications and CMOS-based inverters. Given that these devices belong to distinct fields with unique challenges and research objectives, the following three subsections provide a brief analysis of each area. More detailed discussion please refer to the corresponding chapters of this thesis outlined in Section 1.5.

1.3.1 Microwave Filter

The design of microwave filters can be broadly divided into three steps: topology synthesis, physical dimensioning, and design optimization. While the first two steps are straightforward and relatively simple to automate, design optimization is the most critical and challenging step that requires much more attention. Given an initial design, i.e., an initial 3D structure of a filter constructed by physical dimensioning, design optimization aims to produce the final practical implementation, ready for fabrication and meeting all stringent specifications. Despite several methods proposed in recent decades, there are still some issues hindering the implementation of a fully automated design methodology. Specifically,

- Most filter optimization methods are proposed and validated only on direct-coupled filters without transmission zeros or other specific types. There is still a lack of methods designed for more general types of filters.
- Most filter optimization methods still need human interaction and decision-making to escape local optima or even involve manual preparatory work throughout the entire process. Therefore, the success rate is often hard to guarantee.

- There is a lack of methods that can deal with stringent specifications. An entire set of specifications contains requirements for both passband and stopband. When transmission zeros are designed, specifications for the stopband are essential and indispensable, while current proposed methods are deficient in considering this scenario.

From the perspective of optimization algorithms, filter design optimization presents at least two challenges: 1) the landscape characteristic of filter design problems are highly multimodal, making the optimization result very sensitive to the perturbation of design variables; 2) algorithms based on global optimization present acceptable effectiveness in finding required solutions but show low efficiency in many cases. More discussions can be found in Section 3.2 and 3.1.

The research objective is therefore evident: 1) by comprehensively reviewing the published work in filter design optimization, conduct an in-depth understanding of the bottlenecks of design automation; 2) explore the potential path to break the bottlenecks and overcome the above issues; and 3) propose a new methodology that shows efficiency and effectiveness within acceptable runtime for general filter design cases. More discussion can be found in Section 3.3.

1.3.2 Power Amplifier

Designing a power amplifier is not trivial due to its complex circuit matching requirements and the cumbersome simulation process. This challenge becomes even more intricate when designing multistage, wideband PAs with stringent specifications and performance consistency requirements. The current design methodology involves a sequential process from the design of ideal circuits and schematics to layouts, where intensive manual tuning is inevitable. Design automation techniques are therefore expected to revolutionize this process and free people from tedious trial-and-error. However, although some research has been published targeting automated design, there is still a significant gap in achieving design automation. To be specific, the following issues are prominent in the current stage.

- Most published work is proposed only for PAs of a specialized purpose, such as pre-building some reusable passive-component models by machine learning for reuse in the future. There is a lack of work dedicated to a general methodology that can be applied to the majority of PA design problems.

- Most reported work is proposed for designing PAs only at the schematic level without EM simulations. Therefore, these methods can only be applied to simple configurations within sub-6 GHz. When the operating frequency becomes higher, or the configuration of PA becomes complex, the outcome is often hard to guarantee with these reported methods.
- Most published work employs off-the-shelf algorithms that are not specialized for PA design problems. There is a lack of bespoke optimization algorithms that can handle PA design problems with acceptable efficiency. Hence, even without considering the above two limitations, feasible methods often require thousands to tens of thousands of simulation runs, which is unrealistic for practice.

In addition, unlike the design automation of microwave filters, which involves only EM simulation for passive structures, challenge of PA design automation also lies in the complexity of multiple simulation procedures. For instance, when considering the design of monolithic microwave integrated circuit (MMIC) PAs, holistic characterization includes various simulators (e.g., harmonic balance for nonlinear characterization, S-parameter for linear frequency-domain characterization, and momentum for refined EM simulation) and relies on the design kits provided by foundries. Therefore, to correctly design such PAs, it is necessary to build an integrated environment that connects both algorithms and simulation processes. More discussion can be found in Sections 4.2 and 4.1.

In summary, the research objective aims to: 1) comprehensively review the published work in the field of PA design automation and conduct an in-depth understanding of current bottlenecks; 2) explore potential paths to break the bottlenecks and overcome the above issues; and 3) propose a new methodology targeting more general PA structures with reasonable efficiency and effectiveness. More discussion can be found in Section 4.3 and 4.4.

1.3.3 Semiconductor Device

Over the past half-century, the performance improvement of semiconductor devices has primarily benefited from the continuous shrinkage in technology nodes. Designing devices based on a trial-and-error approach was generally feasible. However, as transistors transition from planar to three-dimensional structures, the complexity of processing makes emulation and simulation in advance indispensable. Despite these advancements, in general, TCAD is still in its infancy and there is a lack of algorithmic

design methodology that can be applied to semiconductor devices. This is partially due to the complexity of device simulation, which typically involves multiple physical models and is very computationally expensive. Hence, traditional optimization algorithms are obviously intractable. More analysis of this situation can be found in Section 5.1.

Therefore, the research objective of this topic is to take the first step toward algorithmic design optimization in the semiconductor field. This involves a comprehensive review of existing work in device modeling and design, analyzing key characteristics of device design challenges, proposing suitable algorithms, and validating these through practical applications. For more discussion, please refer to Section 5.3.1 and 5.4.1.

1.4 Contribution and Research Outcomes

Based on the challenges and objectives outlined earlier, this research explored the potential pathway toward machine learning-assisted EDA for distributed-parameter devices/circuits and semiconductor devices. The contribution mainly concentrates on the new design methodologies proposed with specialized optimization algorithms applicable to the corresponding area. To be specific,

- **Microwave Filter** An unsupervised filter design methodology is proposed and validated using two real-world examples. The proposed methodology consists of a systematic sampling method and a two-phase optimization process, each with a bespoke objective function. Design knowledge is comprehensively considered at different stages of design optimization to enhance robustness and improve success rate. A hybrid optimization algorithm incorporating Gaussian process models is proposed to achieve both efficiency and effectiveness. This methodology can handle more general filter specifications rather than specific cases. By incorporating a programmable physical dimensioning method, the design optimization can be completed in approximately half a day on a standard desktop computer without decision and intervention from designers. The outcome of this contribution is published in [J1].

- **Power Amplifier:** A new methodology is proposed for designing MMIC PAs at layout level. By implementing and incorporating an integrated simulation environment, the methodology is compatible with most product design kits and workflows, and is able to conduct required holistic characterization. Bayesian neural

networks are introduced in design optimization to predict and prescreen candidate solutions during optimization. A novel hybrid search strategy is proposed and embedded in global optimization to speed up convergence. Additionally, the effectiveness of the proposed methodology is validated by two MMIC PAs: a 27-31 GHz balanced PA and a 24-31 GHz wideband Doherty PA, with the latter having been taped out for manufacturing. The outcome of this contribution is primarily published in [J2] and [C1].

- **Semiconductor Device:** The contribution in this topic unfolds through two case studies. For the first case study, the structure of the epitaxial layer of an InP pHEMT is optimized to promote terahertz operating frequency. Compared to the commercial pHEMT, the optimized design achieves 57% and 37% improvements in its cut-off frequency and maximum oscillation frequency, respectively, without altering the gate length. For the second case study, the concept of device circuit co-optimization is proposed and validated using CMOS inverters. Using a novel actor-critic-based optimization algorithm, the proposed method achieves better performance on the inverter with planar MOSFETs than with advanced technology. To the best of our knowledge, the algorithmic design method in these studies is proposed for the first time. Additionally, a practical TCAD interface is implemented to form a foundation supporting the above and future research. The outcome of this contribution is published in [J3] and [C2].

The research outcomes include three journal and two conference publications listed as follows:

[J1] **L. Xue**, B. Liu, Y. Yu, Q. S. Cheng, M. Imran, and T. Qiao, "An Unsupervised Microwave Filter Design Optimization Method Based on a Hybrid Surrogate Model-Assisted Evolutionary Algorithm," IEEE Transactions on Microwave Theory Techn., vol. 71, no. 3, pp. 1159–1170, Mar. 2023, doi: 10.1109/TMTT.2022.3219072. **Published**

[J2] B. Liu, **L. Xue**, H. Fan, Y. Ding, M. Imran, and T. Wu, "An Efficient and General Automated Power Amplifier Design Method Based on Surrogate Model Assisted Hybrid Optimization Technique," IEEE Transactions on Microwave Theory Techn. **Accepted**

[J3] **L. Xue**, A. Dixit, N. Kumar, V. Georgiev, and B. Liu, "Machine Learning-Assisted Device Circuit Co-Optimization: A Case Study on Inverter," IEEE Transactions on Electron Devices, vol. 71, no. 12, pp. 7256–7262, Dec. 2024, doi: 10.1109/TED.2024.3476231. **Published**

[C1] **L. Xue**, H. Fan, Y. Ding, and B. Liu, "A Design Methodology of MMIC Power Amplifiers Using AI-driven Design Techniques," in 2023 19th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD), Funchal, Portugal: IEEE, Jul. 2023, pp. 1–4. doi: 10.1109/SMACD58065.2023.10192155. **Published**

[C2] J. Wang, **L. Xue**, B. Liu, and C. Li, "Design of Terahertz InP pHEMT Using Machine Learning Assisted Global Optimization Techniques," in 2021 16th European Microwave Integrated Circuits Conference (EuMIC), London, United Kingdom: IEEE, Apr. 2022, pp. 67–70. doi: 10.23919/EuMIC50153.2022.9784068. **Published**

In all the publications mentioned above, the author (Liyuan Xue) is responsible for identifying the research questions and making major contributions, including proposing the methodology, implementing the interface and algorithm, and conducting the experiments. Note that, in [C2], the pHEMT simulation model was constructed and calibrated by Jing Wang, who also provided a detailed description of the structure of the epitaxial layer. The first author of [J2] is the author's primary supervisor who discussed and proposed the alternate search strategy of the algorithm in Chapter 4. The author (Liyuan Xue) implemented and refined this idea, and integrated the algorithm into the proposed design methodology followed by validation and comparison on two examples.

1.5 Outline of Thesis

There are six chapters in this thesis. This chapter has set out the background and introduced the basic concept of EDA and three research problems with their challenges and objectives. Chapter 2 introduces the fundamental concepts related to optimization algorithms and machine learning techniques used in this thesis, forming the technological foundation for the following chapters. In Chapter 3, microwave filter design automation is thoroughly investigated, including a comprehensive literature review, problem description, explanation of the proposed methodology, and experimental results followed by a summary. Chapter 4 follows a similar structure to Chapter 3, while the research theme is PA design automation. In Chapter 5, an attempt towards algorithmic design optimization for semiconductor devices is explored with two case studies, each including a brief introduction and methodology explanation followed by results and discussion. Finally, Chapter 6 concludes this dissertation and discusses potential future research extensions.

Chapter 2

Optimization Algorithms and Machine Learning Techniques

Chapter 1 provides the background of EDA and outlines the challenges and research objectives across three focused fields. In this chapter, the mathematics and techniques behind the design methodologies proposed in this thesis are explained in detail. The following content is divided into three main sections. First, optimization algorithms are introduced, as they serve as the foundation of most design technologies, including those in this thesis. Next, several supervised learning methods are detailed, along with their implementation. The third section introduces two general frameworks that integrate machine learning with optimization algorithms, forming the backbone of this thesis. Unlike many works in the CAD domain that employ ML techniques independently, as discussed in Section 1.2, the research in this thesis integrates ML models cohesively with optimization algorithms. Further details are provided in the third section. Additionally, this chapter defines and clarifies the majority of symbols and notations used in the following chapters.

2.1 Optimization Algorithms

Optimization algorithms are essential for achieving design automation in almost all aspects. The importance arises from the ability to precisely search for solutions that meet rigorous requirements, i.e., objectives and constraints. While engineers adjust parameters based on their design knowledge, algorithms seek the best solution through their search engine. Therefore, to achieve and even surpass human design abilities,

algorithms must be delicately designed to be robust, effective, and efficient for specific problems. In this section, we first clarify some basic concepts of optimization problems, and then move on to two main categories of search strategies in optimization algorithms: local search methods and heuristics.

2.1.1 Basic Concept

Electronic design problems can often be cast as optimization problems formulated as follows:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & f(\boldsymbol{x}) \\ \text{subject to } & g_i(\boldsymbol{x}) \leq 0 \text{ for } i \in \mathcal{I} \\ & \boldsymbol{x} \in \mathcal{X} \end{aligned} \tag{2.1}$$

where operator $\min_{\boldsymbol{x}}$ indicates a minimization problem, in which $\boldsymbol{x}$ is the design variables within decision domain or design space $\mathcal{X}$, representing the design parameters of a problem. $\mathcal{I}$ is the index set of i. $f(\boldsymbol{x})$ is the objective to be optimized, and $g_i(\boldsymbol{x})$ is the i-th inequality constraints. The aim of the problem is to find the optimal $\boldsymbol{x}$ that minimizes the objective function $f(\boldsymbol{x})$. Additionally, $\mathcal{X}$ is assumed to be a subset of a high-dimensional real domain. In other words, all optimization problems discussed hereinafter are considered continuous.

Mathematically speaking, Equation (2.1) is the canonical single-objective constrained optimization [53], whereas Equation (2.2) is more appropriate for expressing a practical design problem considered in this thesis:

$$\begin{aligned} \operatorname*{argmin}_{\boldsymbol{x}} \quad & f(\mathcal{R}(\boldsymbol{x}), \omega) \\ \text{subject to } & g_i(\mathcal{R}(\boldsymbol{x}), \omega) \leq 0 \text{ for } i \in \mathcal{I} \\ & \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{c} \\ & \boldsymbol{x} \in [x_L, x_H]^d \\ & \omega \in [\omega_L, \omega_H] \\ & \mathcal{R}(\boldsymbol{x}) \text{is a computational-expensive function} \end{aligned} \tag{2.2}$$

where the operator $\operatorname{argmin}_{\boldsymbol{x}}$ refers to the value of the variable $\boldsymbol{x}$ that minimizes a given function. $\mathcal{R}(\boldsymbol{x})$ is the computationally expensive function often performed by commercial simulation software. $f(\,\cdot\,, \omega)$ represents the objective function with an adjoint variable ω corresponding to some independent parameters within interval $[\omega_L, \omega_H]$, e.g.

frequency range. $\boldsymbol{Ax} \leq \boldsymbol{c}$ describes the geometry constraints through linear inequality. The design parameters $\boldsymbol{x}$ are bounded by a hypercubic design space denoted by $[x_L, x_H]^d$ of d dimension. Unless stated otherwise, all optimization problems in this thesis are considered or converted to minimization problems.

Item	**Feature**
Maximization	Output Power (P_{out})
Frequency (ω) Range	21 - 25 GHz
Specification 1	Gain (G) $\geq$ 20 dB
Specification 2	Efficiency (Eff) $\geq$ 30%
Specification 3	$S_{11} \leq$ -15 dB
Simulation	EM and circuit simulation
Search Range	$[x_L, x_H]^d$

Table 2.1: Example of a power amplifier design problem

$$
\begin{aligned}
\underset{\boldsymbol{x}}{\text{argmin}} \quad & P_{\text{out}}^{\text{ref}} - \min_{\omega}(P_{\text{out}}(\boldsymbol{x}, \omega)) \\
\text{subject to} \quad & \min_{\omega}(G(\boldsymbol{x}, \omega)) \geq 20 \text{ dB} \\
& \min_{\omega}(Eff(\boldsymbol{x}, \omega)) \geq 30\% \\
& \max_{\omega}(S_{11}(\boldsymbol{x}, \omega)) \leq -15 \text{ dB} \\
& \boldsymbol{x} \in [x_L, x_H]^d \\
& \omega \in [21, 25] \text{ GHz}
\end{aligned}
\tag{2.3}
$$

Table 2.1 provides an example of a power amplifier design problem and shows how the performance requirements (i.e., specifications with notations in bracket) are converted to an optimization problem as formulated in Equation (2.3). $\min_{\omega}(\cdot)$ and $\max_{\omega}(\cdot)$ represent the minimum and maximum values of a given function over frequency ω. Notation S_{11} in the table denotes the input reflection coefficient in dB of the circuit, and $P_{\text{out}}^{\text{ref}}$ means the reference output power.

When there is no objective to optimize and only constraints to satisfy—a situation that often occurs when the performance requirements are so stringent that no margin is predictably available, Equation (2.1) degrades to a feasibility problem [54]. The aim of a feasibility problem is to find a feasible solution that satisfies all constraints. Although the feasibility problem involves slightly different mathematical theory compared to optimization, it is not distinguished from an engineering perspective and instead is viewed

as a special case of Equation (2.1). Table 2.2 provides an example of a microwave filter design problem which can be cast as a feasibility problem. As the same notation in Table 2.1, S_{11} and S_{21} denote S-parameters of the microwave filter, representing the reflection and transmission coefficient, respectively. Equation (2.4) shows the formulated problem.

Item	**Feature**
Specification 1	$S_{11} \leq -20$ dB in 5 - 6 GHz
Specification 2	$S_{21} \leq -20$ dB in 4 – 4.8 GHz
Specification 3	$S_{21} \leq -20$ dB in 6.2 – 7 GHz
Simulation	EM simulation
Search Range	$[x_L, x_H]^d$

Table 2.2: Examples of microwave filter design problem

$$
\begin{aligned}
\text{find} \quad & \boldsymbol{x} \\
\text{subject to} \quad & \max_{\omega \in \Omega_1}(S_{11}(\boldsymbol{x}, \omega)) \leq -20 \text{ dB} \\
& \max_{\omega \in \Omega_2}(S_{21}(\boldsymbol{x}, \omega)) \leq -20 \text{ dB} \\
& \boldsymbol{x} \in [x_L, x_H]^d \\
& \Omega_1 = [5, 6] \text{ GHz} \\
& \Omega_2 = [4, 4.8] \cup [6.2, 7] \text{ GHz}
\end{aligned}
\tag{2.4}
$$

Before a problem is ready to be solved or optimized, one should consider the approach to handling constraints, as optimization engines are often blind to constraints. This issue depends on the practical problem and the difficulty or stringency of the constraints. The most commonly used method involves static or adaptive penalty functions (or fitness functions in some context) formulated by

$$
F(\boldsymbol{x}) = f(\boldsymbol{x}) + \sum_{i \in \mathcal{I}} \alpha_i \max(g_i(\boldsymbol{x}), 0) \tag{2.5}
$$

where the parameters α_i are the predefined weighting coefficients and are kept fixed for the static case during optimization, and function $\max(\cdot, 0)$ compares the given value with 0 and outputs the larger one.

Additionally, optimization problems formulated in Equation (2.2) are also known as expensive optimization (EO) [55, 56] or black-box optimization (BBO) [57, 58], where their goal is to find a promising solution within a limited evaluation budget. EO and BBO are common in the real world and are frequently employed in many complex scenarios beyond design automation, such as hyperparameter tuning, financial trading optimization, and chemical process optimization.

2.1.2 Local Search Method: Nelder-Mead simplex

If both objective $f(\boldsymbol{x})$ and constraints $g(\boldsymbol{x})$ in Equation (2.1) are analytic or straightforward to evaluate, several methods can be employed to solve the problem, at least in terms of finding local optima. Examples of such methods include gradient (i.e., the derivative vector of a multivariate function) descent and Newton-like methods. These methods iteratively move the solution towards the steepest descent, or the direction of negative gradient, thereby reducing the objective function value. However, when a computationally expensive function is involved, evaluating the objective or constraints becomes costly, making the gradient difficult, or even impossible, to obtain. This necessitates the use of derivative-free methods to alleviate the function evaluation burden, such as Powell's method [59], the Nelder-Mead (NM) simplex method [60], and the Trust-region (TR) method. In this section, the Nelder-Mead simplex method is primarily introduced.

The NM simplex method is a widely used derivative-free local search algorithm that is suitable for non-smooth or even discontinuous landscapes. It is known to converge quickly with a relatively small number of function evaluations, thus achieving preferable results within an acceptable cost. Due to its robustness, effectiveness, and ease of use, it is employed as the local search engine in microwave filter design automation described in Chapter 3.

The pseudo-code of NM simplex method is shown in Algorithm 1 (denoted by `NMSimplex`$(\cdot)$ hereafter). It starts from a set of $d+1$ points (i.e., the initial simplex) that lie in different hyperplanes (i.e., the so-called nondegenerate working simplex), where d refers to the dimension of the design variable. Then four primary operations consisting of *reflection*, *expansion*, *contraction*, and *shrinkage* are performed iteratively according

Algorithm 1 Nelder-Mead Simplex Method (`NMSimplex`$(\cdot)$)

Input: Initial solution $\boldsymbol{x}_{\text{ini}} \in \mathbb{R}^d$, objective function function $f(\cdot)$.

$\mathcal{X} \leftarrow \{\boldsymbol{x}_i \mid \boldsymbol{x}_i = \boldsymbol{x}_{\text{ini}} + \epsilon\boldsymbol{\delta}_i, i = 0, \ldots, d\}$ ▷ Generate initial simplex.
repeat
  Order simplex $\mathcal{S}$ by $f(\cdot)$, so that $f(\boldsymbol{x}_0) \leq f(\boldsymbol{x}_1) \leq \cdots \leq f(\boldsymbol{x}_d)$
  $\boldsymbol{m} \leftarrow \sum_{i=0}^{d-1} \boldsymbol{x}_i/d$ ▷ Compute centroid of top $d-1$ points.
  $\boldsymbol{x}_r \leftarrow 2\boldsymbol{m} - \boldsymbol{x}_d$ ▷ Reflection.
  if $f(\boldsymbol{x}_0) \leq f(\boldsymbol{x}_r) < f(\boldsymbol{x}_{d-1})$ **then**
    $\boldsymbol{x}_d \leftarrow \boldsymbol{x}_r$ and **continue**
  end if
  if $f(\boldsymbol{x}_r) < f(\boldsymbol{x}_0)$ **then**
    $\boldsymbol{x}_s \leftarrow \boldsymbol{m} + 2(\boldsymbol{m} - \boldsymbol{x}_d)$ ▷ Expansion.
    if $f(\boldsymbol{x}_s) < f(\boldsymbol{x}_r)$ **then**
      $\boldsymbol{x}_d \leftarrow \boldsymbol{x}_s$ and **continue**
    else
      $\boldsymbol{x}_d \leftarrow \boldsymbol{x}_r$ and **continue**
    end if
  end if
  if $f(\boldsymbol{x}_{d-1}) \leq f(\boldsymbol{x}_r) < f(\boldsymbol{x}_d)$ **then**
    $\boldsymbol{x}_c \leftarrow \boldsymbol{m} + (\boldsymbol{x}_r - \boldsymbol{m})/2$ ▷ Outside Contraction.
    if $f(\boldsymbol{x}_c) < f(\boldsymbol{x}_r)$ **then**
      $\boldsymbol{x}_d \leftarrow \boldsymbol{x}_c$ and **continue**
    else
      break
    end if
  end if
  if $f(\boldsymbol{x}_r) \geq f(\boldsymbol{x}_d)$ **then**
    $\boldsymbol{x}_{cc} \leftarrow \boldsymbol{m} + (\boldsymbol{x}_d - \boldsymbol{m})/2$ ▷ Inside Contraction.
    if $f(\boldsymbol{x}_{cc}) < f(\boldsymbol{x}_d)$ **then**
      $\boldsymbol{x}_d \leftarrow \boldsymbol{x}_{cc}$ and **continue**
    else
      break
    end if
  end if
  for $i \leftarrow 1$ **to** d **do**
    $\boldsymbol{v}_i = \boldsymbol{x}_0 + (\boldsymbol{x}_i - \boldsymbol{x}_0)/2$ ▷ Shrinkage.
  end for
  $\mathcal{X} \leftarrow \{\boldsymbol{x}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_d\}$
until Stopping criteria are satisfied

Output: Best solution $\boldsymbol{x}_0$ and the corresponding function value $f(\boldsymbol{x}_0)$

to the different conditions depending on the comparison of function values. In each iteration, at least one solution is updated, and no more than two new solutions are evaluated by the given function. The stopping criteria can be the exhaustion of the number of function evaluations or the achievement of desired solution accuracy.

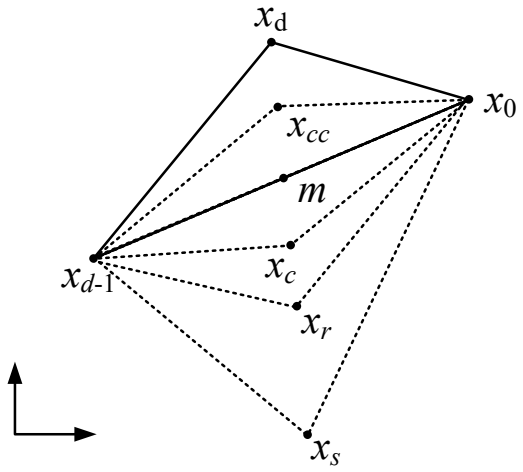


Figure 2.1: Illustration of the solutions in NM simplex method [61]

NM simplex method is straightforward to understand and easy to use. The reflection, expansion, and contraction operations are used to explore possible directions for improving adaptively, while shrinkage is employed to zoom in on the current search region and further exploit when the current step size (i.e. the scale of the simplex) is no longer effective. Figure 2.1 illustrates the relationship among different solutions on a two-dimensional plane during optimization. By setting the worst point as a fulcrum, this method can gradually converge in the desired direction to decrease the function value. For more details on its convergence properties, one can refer to [60].

2.1.3 Evolutionary Algorithm: Differential Evolution

The NM simplex method is capable of performing local searches within the design space of an EO or BBO problem; however, it is limited in finding global solutions in most cases. In this subsection, a global optimization method is introduced, namely Evolutionary algorithms (EAs). EAs, which are inspired by biological evolution and the characteristics of live organisms, are popular for solving problems formulated as BBO in Equation (2.1). EAs commonly include algorithms like genetic algorithm (GA), differential evolution (DE), particle swarm optimization (PSO), and ant colony algorithm (ACO). Inspired by the concept of "survival of the fittest" from natural evolution, EAs iteratively generate new individuals using specific evolutionary operators and select those with higher fitness to advance to the next generation. EAs can efficiently find satisfactory solutions without requiring gradient information, making them highly suitable

for solving real-world problems. Compared with traditional methods, such as Newton's method, EAs are capable of solving non-convex, discontinuous, and non-differentiable problems. In contrast to the NM method introduced in the last subsection, EAs are also capable of conducting global search, aiming at finding the global optima.

Among the different EAs, the DE algorithm is often the first choice when the design variable is in the real domain. It is based on the differences between design vectors and is very straightforward to use with only a few control parameters, including the scaling factor F and crossover rate CR. The DE algorithm has good convergence properties and surpasses many other algorithms in complex benchmark problems [62, 63]. It was therefore selected as the global search engine in this thesis, detailed as follows.

Algorithm 2 Differential Evolution Algorithm

Input: Objective function $f(\cdot)$, population size N, corssover rate CR, scaling factor F.

$\mathcal{P} \leftarrow \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ ▷ Population initialization
repeat
  for each $\boldsymbol{x}_i \in \mathcal{P}$ **do**
    $\boldsymbol{x}_n \leftarrow \boldsymbol{x}_i$
    $r_1, r_2, r_3 \leftarrow \texttt{randidx}(N)$
    $\boldsymbol{v}_i \leftarrow \boldsymbol{x}_{r_1} + F \cdot (\boldsymbol{x}_{r_2} - \boldsymbol{x}_{r_3})$ ▷ Mutation
    for $j \leftarrow 1$ **to** d **do**
      if $\texttt{rand}() \leq CR \mid \texttt{randidx}(d) = i$ **then**
        $x_n^j \leftarrow v_i^j$ ▷ Crossover
      end if
    end for
    if $f(\boldsymbol{x}_n) < f(\boldsymbol{x}_i)$ **then**
      $\boldsymbol{x}_i \leftarrow \boldsymbol{x}_n$ ▷ Selection
    end if
  end for
until Stopping criteria are satisfied

Output: Best solution and the corresponding function value

The pseudo-code of the DE algorithm employed in the following work is shown in Algorithm 2, where the function $\texttt{rand}(\cdot)$ produces a random number from a uniform distribution within [0,1], and the function $\texttt{randidx}(\cdot)$ outputs non-duplicate indexes within the given number. The DE algorithm primarily encompasses three operations: *mutation*, *crossover*, and *selection*. It begins with the initialization of a population consisting of N individuals denoted by $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, where each one represents a potential solution in the search space. During the mutation step, a mutant vector $\boldsymbol{v}_i$ is generated by first randomly selecting three different individuals and then computing with a predefined strategy. In Algorithm 2, the mutation operation is performed by adding the

scaled vector difference of two randomly selected vectors $\boldsymbol{x}_{r_2}$ and $\boldsymbol{x}_{r_3}$ to another one $\boldsymbol{x}_{r_1}$. Subsequently, for each dimension of the mutant vector, algorithm decides whether to take the value from the mutant vector or the original one by comparing the crossover rate with a random number—the process known as crossover. As for the selection step, function value of the new solution is evaluated and compared with the existing one, and only the better is retained in the population. The main loop continues until stopping criteria, such as a maximum number of generations or a satisfactory fitness level, are met.

Apart from the hyperparameters F and CR, there are also several different mutation strategies that can balance exploration (i.e., the ability to explore a wider region within the design space) and exploitation (i.e., the ability to refine the current promising region) capabilities of the algorithm, resulting in different DE variants. The mutation strategy formulated in Algorithm 2 is called DE/rand/1, which aims to maintain high diversity within the population, largely exploring the design space. A more moderate version is formulated in Equation (2.6), called DE/current-to-best/1. This strategy perturbs the current solution, rather than a random one, and also incorporates the feature from the best solution $\boldsymbol{x}_{r_{best}}$. This helps spread favorable patterns from the best solution and often converges more quickly, especially when the problem requires less exploration ability. Equation (2.7) formulates the most aggressive mutation strategy, called DE/best/1, which widely disseminates the feature of the current best solution to each individual, regardless of the current selected one. This strategy is efficient in exploitation but is at risk of being trapped in a local optimum, especially when the global optimum lies in a narrow valley (as is often the case in microwave filter design problems explained in the next chapter).

$$\boldsymbol{v}_i \leftarrow \boldsymbol{x}_{r_i} + F \cdot (\boldsymbol{x}_{r_{best}} - \boldsymbol{x}_{r_i}) + F \cdot (\boldsymbol{x}_{r_2} - \boldsymbol{x}_{r_3}) \tag{2.6}$$

$$\boldsymbol{v}_i \leftarrow \boldsymbol{x}_{r_{best}} + F \cdot (\boldsymbol{x}_{r_1} - \boldsymbol{x}_{r_2}) \tag{2.7}$$

DE algorithm is efficient in solving real domain problems with effective global optimization capability. However, each iteration requires N evaluations of the objective function as list in Line 12 of Algorithm 2, making it difficult to apply directly to computationally expensive problems. Typically, DE algorithm requires thousands to tens of thousands of evaluations to converge (or find a satisfactory solution) when the design variable has tens of dimensions. This is often unacceptable for many real-world problems, particularly the design automation problems considered in this thesis, where each function evaluation can take several minutes to tens of minutes, potentially extending

the entire optimization process to over half a month. In Section 2.3, the introduction of two optimization frameworks, namely Bayesian optimization (BO) and surrogate model-assisted evolutionary algorithms (SAEAs), will show how this issue can be alleviated by incorporating machine learning techniques. But before that, several machine learning techniques will be introduced.

2.2 Supervised Learning

As the most predominant and substantial technique in machine learning, supervised learning has attracted so much attention due to its effectiveness and versatility in solving a wide range of problems. This section introduces the basic concept of supervised learning followed by a discussion about three cutting-edge supervised learning techniques: Gaussian process (GP), deep neural network (DNN), and Bayesian neural network (BNN), which form the ML foundation for the following chapters.

2.2.1 Basic Concept

Given a training set of N input-output pairs $\{(\boldsymbol{x}_i, y_i) \mid i = 1, \ldots, N\}$, supervised learning tries to construct a model $\mathcal{M}_\phi(\cdot)$ to mimic the unknown (or known but complex to explicitly illustrate) behavior or relationship between $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$, parameterized by ϕ. In this context, y_i (assume a scalar for simplicity) is also called the ground truth—the true value ones hope the model will predict—and the formulation of $\mathcal{M}_\phi(\cdot)$ is called hypothesis [26]. Supervised learning assumes that the training set consists of a sample of independent and identically distributed pairs. By defining and minimizing the loss function $L(\mathcal{M}_\phi(\boldsymbol{x}_i), y_i)$ on the training set, such as the mean square error function, the trained model is expected to generalize and predict on unseen data.

However, beyond the basic assumption of training data, supervised learning faces the bias-variance issue. Imagine a trained model that has a small bias on training data but exhibits high variance for a particular input. This indicates that the model is overfitting or "too flexible" on the training set and may not generalize well to unseen data. A more detailed discussion is illustrated in Table 2.3, where the upper left corner represents the

ideal state—low bias and low variance—while the lower right corner represents a state to avoid. When variance is small but bias is high, one should consider improving the model's capability (e.g., by adding more terms or increasing the number of parameters in $\mathcal{M}_\phi(\cdot)$). Generally, there is often a tradeoff between bias and variance.

Table 2.3: Bias-Variance tradeoff

		Variance	
		Small	**High**
Bias	**Small**	Good model	Overfitting
	High	Underfitting	Bad model

The simplest supervised learning is linear regression, which assumes a linear relationship between $\boldsymbol{x}$ and y. By setting the sum of squared errors as the loss function, linear regression can be solved in closed form using the least squares method. However, linear regression lacks the capability of modeling complex landscapes, making it less suitable for many engineering problems. Therefore, more advanced models, such as Gaussian processes and deep neural networks, should be introduced.

2.2.2 Gaussian Process Model

Gaussian process model is a widely used supervised learning method in engineering optimization, whose strengths include its strong learning and characterization capability and the ability to provide a statistically grounded prediction uncertainty. The GP model treats the training data $\boldsymbol{y}$ as a set of N samples from a multivariate Gaussian distribution. Therefore, the likelihood function can be expressed in terms of samples $\boldsymbol{y}$ as

$$L_{\mathrm{GP}} = \frac{1}{\left(2\pi\sigma^2\right)^{N/2}\left|\boldsymbol{R}\right|^{1/2}} \exp\left[-\frac{(\boldsymbol{y}-\mathbf{1}\mu)^{\mathrm{T}}\boldsymbol{R}^{-1}(\boldsymbol{y}-\mathbf{1}\mu)}{2\sigma^2}\right] \tag{2.8}$$

where μ and σ^2 are the mean and variance of the Gaussian process model. $\mathbf{1}$ is a $N \times 1$ vector of ones, $\boldsymbol{R}$ is the $N \times N$ covariance matrix defined by the correlation function (i.e., the Gaussian kernel function)

$$R_{i,j} = \mathrm{Corr}\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) = \exp\left(-\sum_{l=1}^{d} \theta_l \left|x_i^l - x_j^l\right|^{p_l}\right),\ \theta_l > 0, 1 \le p_l \le 2 \tag{2.9}$$

where d is the dimension of $\boldsymbol{x}$, different samples are indicated by i and j, and $\boldsymbol{\theta}$ and $\boldsymbol{p}$ are hyperparameters describing how fast the correlation decreases on the l-th variable and the corresponding function smoothness, respectively. For a given set of training data, the maximum likelihood (Equation (2.8)) estimates the model parameters that maximize the probability of observed samples. Therefore, by setting $\frac{\partial}{\partial \mu}\log(L_{\mathrm{GP}})$ and $\frac{\partial}{\partial \sigma^2}\log(L_{\mathrm{GP}})$ to zero, and assuming the hyperparameter $\boldsymbol{\theta}$ and $\boldsymbol{p}$ are known, the μ and σ in Equation (2.8) can be obtained in a closed form, where

$$\begin{aligned}\hat{\mu} &= \frac{\mathbf{1}^T \boldsymbol{R}^{-1}\boldsymbol{y}}{(\mathbf{1}^T \boldsymbol{R}^{-1}\mathbf{1})^{-1}} \\ \hat{\sigma} &= \frac{(\boldsymbol{y}-\mathbf{1}\hat{\mu})^T \boldsymbol{R}^{-1}(\boldsymbol{y}-\mathbf{1}\hat{\mu})}{N}\end{aligned} \tag{2.10}$$

Substituting Equation (2.10) into (2.8), the likelihood function can be maximized numerically either by Quasi-Netwon method or others, obtaining the optimal hyperparameter values formulated by

$$\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{p}} = \arg\max_{\boldsymbol{\theta},\boldsymbol{p}} \left(-\frac{N}{2}\ln\hat{\sigma}^2 - \frac{1}{2}\ln|\boldsymbol{R}| \right) \tag{2.11}$$

Given a new design $\boldsymbol{x}^*$, the prediction value $\hat{y}(\boldsymbol{x}^*)$ and the uncertainty $\hat{s}(\boldsymbol{x}^*)$ can be obtained by best linear unbiased estimation and mean square error:

$$\hat{y}(\boldsymbol{x}^*) = \hat{\mu} + \boldsymbol{r}^T \boldsymbol{R}(\boldsymbol{y}-\mathbf{1}\hat{\mu}) \tag{2.12}$$

$$\hat{s}^2(\boldsymbol{x}^*) = \hat{\sigma}^2 \left[1 - \boldsymbol{r}^T \boldsymbol{R}^{-1}\boldsymbol{r} + \frac{\left(1 - \mathbf{1}^T \boldsymbol{R}^{-1}\boldsymbol{r}\right)^2}{(\mathbf{1}^T \boldsymbol{R}^{-1}\mathbf{1})} \right] \tag{2.13}$$

where $\boldsymbol{r} = [\mathrm{Corr}(\boldsymbol{x}^*, \boldsymbol{x}_1), \mathrm{Corr}(\boldsymbol{x}^*, \boldsymbol{x}_2), ..., \mathrm{Corr}(\boldsymbol{x}^*, \boldsymbol{x}_N)]^T$ describing the correlation between $\boldsymbol{x}^*$ and all sample designs. More details about GP can be found in [64].

The GP model has advantages in terms of its preciseness and tractability. As a non-parametric model [65], the number of parameters of a GP model grows with the size of the observed dataset (i.e., the training data). When a moderate-sized training set is given, GP modeling is a theoretically principled method with a relatively small number of hyperparameters to determine, making it ideal for small-sample modeling. However, training a GP model suffers from high computational complexity, specifically $O(N^3 d)$, where N is the size of the training set and d is the dimension of the input variables

$\boldsymbol{x}$. Given that N is at least linearly dependent on d, the computational complexity approaches the fourth power of d. Additionally, considering the optimization iterations required to solve Equation (2.11), the time consumption of applying GP models needs serious consideration.

Moreover, GPs can only model cases where the output is scalar. When the output variable is a vector, i.e., with multiple values, GP modeling cannot be directly applied. A remedy is to construct several models for each output element, although this approach ignores the correlation between output variables. In such scenarios, a more appropriate method is to use deep neural networks, which can directly model multiple input and output variables comprehensively.

2.2.3 Deep Neural Network

Neural networks have emerged in recent years as one of the most powerful techniques for practical application, as introduced in Chapter 1. Inspired by how the human brain processes information—though the paradigm has now largely shifted away from biological inspiration, a deep neural network consists of a stack of several functional layers. These layers, along with the connections between them, mimic the neurons and axons of the human brain, passing and processing information to extract abstract representations that produce an output. While the first artificial neural networks were proposed [66], their true potential was gradually discovered with the advent of powerful computational resources and large datasets. Various different configurations of neurons have been proposed to drive advancements in different fields.

In this thesis, a kind of neural network called multilayer perceptron (MLP) is employed as a regression method to model the behavior of multivariant input-output pairs. It involves three operations (functional layers): linear combination, nonlinear activation, and batch normalization. The linear combination and nonlinear activation can be formulated as follows

$$\boldsymbol{z} = h\left(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}\right) \tag{2.14}$$

where $h(\cdot)$ is the activation function, $\boldsymbol{W}$ and $\boldsymbol{b}$ are trainable parameters, and $\boldsymbol{z}$ is the output after these operations. To facilitate the subsequent derivation, element form of Equation (2.14) is formulated as:

$$z_j = h\left(\sum_{i=1}^{d} w_{ij}x_i + b_j\right),\ j = 1, \ldots, M \tag{2.15}$$

where i and j index the input dimension d for $\boldsymbol{x}$ and the output dimension M for $\boldsymbol{z}$, respectively. The linear combination and nonlinear activation transform the input data to a higher-dimensional space for feature extraction; therefore, M is always greater than d.

For each output neuron z_j, given a training batch $\mathcal{B}$ (a subset of the given training set), the corresponding output value is denoted by $z_{j;k}$, where $k \in \{1, ..., |\mathcal{B}|\}$. Hence, for normalization operation [67], let

$$\begin{aligned}
\mu_j &= \frac{1}{|\mathcal{B}|}\sum_{k=1}^{|\mathcal{B}|} z_{j;k} \\
\sigma_j^2 &= \frac{1}{|\mathcal{B}|}\sum_{k=1}^{|\mathcal{B}|} \left(z_{j;k} - \mu_j\right)^2 \\
\hat{z}_{j;k} &= \frac{z_{j;k} - \mu_j}{\sqrt{\sigma_j^2 + \epsilon}} \\
y_{j;k} &= \gamma_j z_{j;k} + \psi_j
\end{aligned} \tag{2.16}$$

where γ_j and ψ_j are trainable parameters, μ_j and σ_j^2 are statistical information computed based on the current training batch, and ϵ is a small number (often 1×10^{-5}) for numerical stability. Equation (2.16) normalizes the output from activation to the normal distribution (zero mean and unit variance), then scales and shifts the result through trainable parameters. This process is known as batch normalization [68]. It offers benefits by accelerating the training of neural networks and introducing a slight regularization effect, helping the model generalizes better. Furthermore, by applying batch normalization on an element-wise basis of input data, the input variables are standardized, effectively disregarding differences in scales or magnitudes——a common situation encountered in engineering optimization.

The activation function used in this thesis includes $\tanh(\cdot)$ and $\text{relu}(\cdot)$, defined by

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{2.17}$$

$$\text{relu}(x) = \max(0, x) \tag{2.18}$$

where $\max(0, \cdot)$ output the maximum value compared to 0. Additionaly, when the output variable is bound within $[0, 1]$, sigmoid function is also used, defined by

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}} \tag{2.19}$$

Stacking (2.14) and (2.16) multiple times creates deeper neural networks. After defining the loss function at the output, the network's parameters can be trained using back-propagation [69], which iteratively applies the chain rule to each calculation. Assuming $y_{;k}$ is the output of the neural network, ignoring neuron index j in derivation, and the loss function is denoted by L. Hence,

$$\begin{aligned}
\frac{\partial L}{\partial \hat{z}_{;k}} &= \frac{\partial L}{\partial y_{;k}} \cdot \gamma \\
\frac{\partial L}{\partial \sigma} &= \sum_{k=1}^{|\mathcal{B}|} \frac{\partial L}{\partial \widehat{z}_{;k}} \cdot (z_{;k} - \mu) \cdot \frac{-1}{2} \left(\sigma^2 + \epsilon\right)^{-3/2} \\
\frac{\partial L}{\partial \mu} &= \left(\sum_{k=1}^{|\mathcal{B}|} \frac{\partial L}{\partial \widehat{z}_{;k}} \cdot \frac{-1}{\sqrt{\sigma^2 + \epsilon}} \right) + \frac{\partial L}{\partial \sigma^2} \cdot \frac{\sum_{k=1}^{|\mathcal{B}|} -2\,(z_{;k} - \mu)}{|\mathcal{B}|} \\
\frac{\partial L}{\partial z_{;k}} &= \frac{\partial L}{\partial \widehat{z}_{;k}} \cdot \frac{1}{\sqrt{\sigma^2 + \epsilon}} + \frac{\partial L}{\partial \sigma^2} \cdot \frac{2\,(z_{;k} - \mu)}{|\mathcal{B}|} + \frac{\partial L}{\partial \mu} \cdot \frac{1}{|\mathcal{B}|} \\
\frac{\partial L}{\partial \gamma} &= \sum_{k=1}^{|\mathcal{B}|} \frac{\partial L}{\partial y_{;k}} \cdot \widehat{z}_{;k} \\
\frac{\partial L}{\partial \psi} &= \sum_{k=1}^{|\mathcal{B}|} \frac{\partial L}{\partial y_{;k}}
\end{aligned} \tag{2.20}$$

Equations in (2.20) indicate the fully differentiable properties of the neural network, allowing the model's parameters to be trained using stochastic gradient descent (SGD) or ADAM [70]. Additionally, during the training procedure, the data is often divided into two sets: the training set and the test set. This division helps monitor the decrease in the loss function and prevent overfitting. Due to the powerful approximation

capabilities of neural networks, guaranteed by universal approximation theorems [71], overfitting can often be inevitable. The simplest and most commonly used approach to prevent overfitting is early stopping, where the stopping point is determined by setting a threshold for the reduction rate of the loss function.

Figure 2.2 illustrates the differences between GP and DNN models. While DNN models can intrinsically handle multivariate outputs, GP models require separate models for each variable. Therefore, DNNs are more efficient and appropriate for problems with many objectives or constraints for modeling as formulated in Equation (2.2).

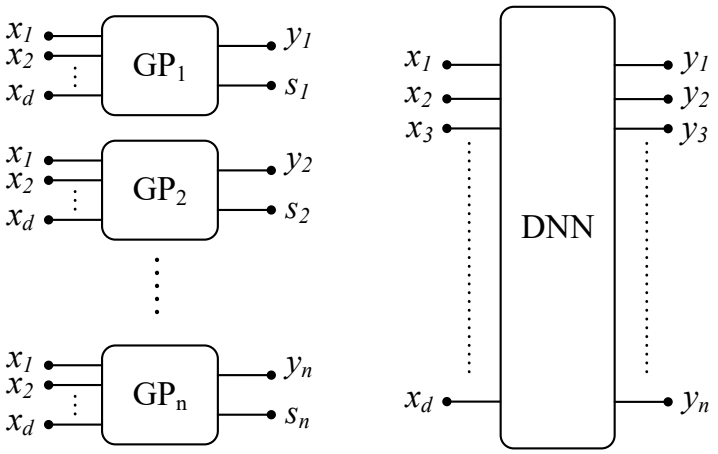


Figure 2.2: Illustration of GP and DNN models for multivariate output.

However, GP models provide predictive uncertainty, which characterizes the confidence or trustworthiness of the prediction—an aspect that is crucial in optimization frameworks introduced in Section 2.3. To enable neural networks to provide predictive uncertainty, an intuitive approach is to replace all the parameters of the network with probability distributions rather than scalar values, thereby transforming it into a probabilistic model. This method is detailed in the next subsection.

2.2.4 Bayesian Neural Network

As shown in Figure 2.3, Bayesian neural networks are the natural extension of the classical neural networks, which not only provide predictions but also quantify uncertainties. Since its introduction, BNNs have sparked widespread attention and stimulated substantial research, particularly in the domain of computational science for engineering [72]. To understand BNNs, it is essential first to recall Bayes' theorem, which states

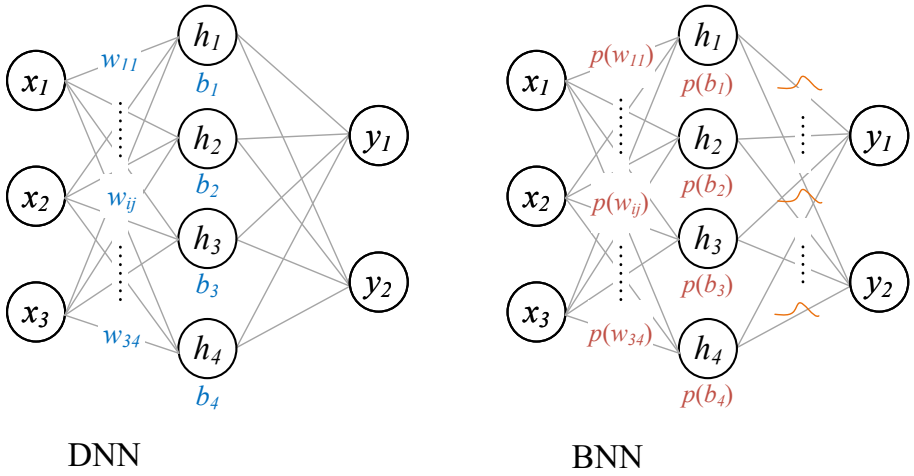


Figure 2.3: Illustration of the structure of DNN and BNN models.

that for two events A and B, the conditional probability $P(A|B)$ of event A occurring while B has occurred is

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{2.21}$$

Equation (2.21) is derived from the product rule of probability, where the probability of events A and B happening simultaneously is $P(A, B) = P(A|B)P(B) = P(B|A)P(A)$. Now, consider a neural network parameterized by $\boldsymbol{w}$, where the prior of $\boldsymbol{w}$ is $p(\boldsymbol{w})$. Given a set of training data $\mathcal{D}$, the posterior distribution of the parameters can be calculated as

$$p(\boldsymbol{w}|\mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{w})p(\boldsymbol{w})}{p(\mathcal{D})} = \frac{p(\mathcal{D}|\boldsymbol{w})p(\boldsymbol{w})}{\int p(\mathcal{D}|\boldsymbol{w})d\boldsymbol{w}} \tag{2.22}$$

The denominator in this formulation is called the marginal likelihood or evidence. Explicitly, Equation (2.23) illustrates the relationship between likelihood, prior, evidence, and posterior, where

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}} \tag{2.23}$$

However, directly computing the posterior distribution is intractable due to the computational burden of the denominator (i.e., the infinite integration). To address this issue, variational inference is employed. A variational distribution $q_{\boldsymbol{\theta}}(\boldsymbol{w})$ is introduced over the parameter set $\boldsymbol{w}$. The parameters of this variational distribution are then adjusted to minimize the dissimilarity between the variational distribution $q_{\boldsymbol{\theta}}(\boldsymbol{w})$ and the true posterior $p(\boldsymbol{w}|\mathcal{D})$, as measured by KL-Divergence:

$$\text{KL}\left[q_{\boldsymbol{\theta}}(\boldsymbol{w})\|p(\boldsymbol{w}|\mathcal{D})\right] = \int q_{\boldsymbol{\theta}}(\boldsymbol{w}) \log \frac{q_{\boldsymbol{\theta}}(\boldsymbol{w})}{p(\boldsymbol{w}|\mathcal{D})} d\boldsymbol{w} \tag{2.24}$$

Equation (2.24) can serve as the objective function for optimization with regards to the variational parameters $\boldsymbol{\theta}$. It can be further simplified as

$$\mathrm{KL}\left[q_{\boldsymbol{\theta}}(\boldsymbol{w})\|p(\boldsymbol{w}|\mathcal{D})\right] = \mathbb{E}_q[\log p(\mathcal{D}|\boldsymbol{w})] - \mathrm{KL}\left[q_{\boldsymbol{\theta}}(\boldsymbol{w})\|p(\boldsymbol{w})\right] + \log p(\mathcal{D}) \tag{2.25}$$

Equation (2.25) can be optimized by a gradient-based solver (i.e., SGD or ADAM as introduced in DNNs). While the last term $\log p(\mathcal{D})$ in Equation (2.25) is constant and does not contribute to gradient for optimization, the remaining terms constitute the well-known evidence lower bound (ELBO). The ELBO consists of maximizing the likelihood estimation $\mathbb{E}_q[\log p(\mathcal{D}|\boldsymbol{w})]$ and the regularization $\mathrm{KL}\left[q_{\boldsymbol{\theta}}(\boldsymbol{w})\|p(\boldsymbol{w})\right]$. In practice, $\mathbb{E}_q[\log p(\mathcal{D}|\boldsymbol{w})]$ is equivalent to the mean square error loss in regression estimated by Monte Carlo sampling, while $\mathrm{KL}\left[q_{\boldsymbol{\theta}}(\boldsymbol{w})\|p(\boldsymbol{w})\right]$ can be computed analytically [73].

Additionally, to make the training of BNN compatible with the backpropagation framework, a trick called reparameterization is introduced. This technique serves as the foundation for pathwise-gradient estimation (i.e., the automatic differentiation frameworks). Considering the calculation of $\mathbb{E}_q[\log p(\mathcal{D}|\boldsymbol{w})]$ in the ELBO, which requires sampling $\boldsymbol{w}$ from its variational distribution $q_{\boldsymbol{\theta}}(\boldsymbol{w})$, define $\boldsymbol{\theta} = \{\boldsymbol{\mu}, \boldsymbol{\sigma}\}$ and let

$$\begin{aligned} \boldsymbol{w} &\sim \mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{\sigma}^2\right) \\ \boldsymbol{w} &= g(\boldsymbol{\theta}, \boldsymbol{\epsilon}) = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon} \end{aligned} \tag{2.26}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ is a sampling set and $\odot$ represents the element-wise product. $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ can then be updated by gradient backpropagation iteratively.

Once the distribution of $p(\boldsymbol{w}|\mathcal{D})$ is obtained, the inference of a trained BNN denoted by $\pi_{\boldsymbol{w}}(\boldsymbol{x})$ is considered as the ensemble of a classic neural network:

$$\hat{\boldsymbol{y}} = \mathop{\mathbb{E}}_{\boldsymbol{w}\sim p(\boldsymbol{w}|\mathcal{D})}\left[\pi_{\boldsymbol{w}}(\boldsymbol{x}^*)\right] \tag{2.27}$$

$$\hat{\boldsymbol{s}}^2 = \mathop{\mathbb{E}}_{\boldsymbol{w}\sim p(\boldsymbol{w}|\mathcal{D})}\left[(\pi_{\boldsymbol{w}}(\boldsymbol{x}^*) - \hat{\boldsymbol{y}})^2\right] \tag{2.28}$$

where Equation (2.27) and (2.28) are the prediction and its corresponding variance (square of pridictive uncertainty).

Compared to GPs, BNNs are more computationally efficient and flexible in capturing multivariant complex patterns. Compared to DNNs, BNNs are more robust in small-sample modeling and provide uncertainty estimates simultaneously. However, the training complexity of BNNs is higher than DNNs, making them potentially overcomplicated for tasks that do not require uncertainty estimation, which is the case for problems discussed in Chapter 5.

2.3 Bayesian Optimization and Surrogate-Assisted Evolutionary Algorithm

Bayesian optimization (BO) and surrogate-assisted evolutionary algorithm (SAEA) are two promising frameworks that offer efficient solutions for solving EO and BBO problems, as discussed in Section 2.1.1. Although, the underlying mechanisms for these two frameworks are similar—both employ machine learning methods to approximate (or surrogate) the objective function while reducing computational cost, BO places greater emphasis on the prior and posterior distribution (Bayes' theorem) of the problem by incorporating specific infill sampling, whereas SAEA focuses primarily on EA strategies and the incorporation of surrogate models. To compare the similarities and differences between these two frameworks, the pseudo-codes of them are shown in Algorithm 3 and 4.

Algorithm 3 Bayesian optimization (BO)

Input: Objective function $f(\boldsymbol{x})$, acquisition function $u(\cdot)$
1: Initialize the observed dataset $\mathcal{D}$ and evaluate each sample by $f(\boldsymbol{x})$.
2: Set prior to the probabilistic model $\mathcal{M}(\cdot)$
3: **repeat**
4: Fit the posterior of $\mathcal{M}(\cdot)$ to $\mathcal{D}$
5: Optimize acquisition function over $\mathcal{M}(\cdot)$ to get the next solution $\boldsymbol{x}_n$
6: Evaluate $\boldsymbol{x}_n$ to obtain $f(\boldsymbol{x}_n)$
7: Add $(\boldsymbol{x}_n, f(\boldsymbol{x}_n))$ to $\mathcal{D}$
8: **until** Stopping criteria are satisfied
Output: Best solution and the corresponding function value

BO begins with the initialization of an observed dataset, where each sample is evaluated. Following this, a probabilistic model, often a Gaussian process model, is established with a prior distribution. In the main loop, the model is first fitted by computing posterior distribution over the current observed dataset $\mathcal{D}$. Then the acquisition function

$u(\cdot)$ is optimized over the model to determine the next solution $\boldsymbol{x}_n$. Specifically, for a minimization problem, this process is formulated by

$$\boldsymbol{x}_n = \underset{\boldsymbol{x}\in\mathcal{X}}{\operatorname{argmin}}\, u(\mathcal{M}(\boldsymbol{x})) \tag{2.29}$$

The acquisition function, also known as infill criteria, aims to reward solutions with risk (i.e., uncertainty) while balancing exploration (searching new region of the design space) and exploitation (refining the search in promising region) to efficiently navigate the search space, so-called infill sampling. The optimization of Equation (2.29) is often performed by gride search due to its lower computational cost compared to the original problem. Once some new solutions are selected, they are evaluated to obtain the ground truth and then added to $\mathcal{D}$. This loop continues until some stopping criteria are met, such as the time limit or the maximum number of iterations. BO has been proven to have global optimization capabilities [74, 75], making it particularly advantageous for solving EO and BBO problems with global optima.

Algorithm 4 Surrogate-Assisted Evolutionary Algorithm (SAEA)

Input: Objective function $f(\boldsymbol{x})$, population size N
1: Initialize the population $\mathcal{P}$ of N individuals
2: Evaluate each individual and form training dataset $\mathcal{D}$
3: **repeat**
4: Train surrogate model $\mathcal{M}(\cdot)$ by $\mathcal{D}$
5: Generate offspring $\mathcal{P}_o$ by applying mutation and crossover operations
6: Predict and prescreen solutions in $\mathcal{P}_o$ by $\mathcal{M}(\cdot)$
7: Select a small subset $\mathcal{S} \subset \mathcal{P}_o$ for real evaluation by $f(\boldsymbol{x})$
8: Combine evaluated solutions in $\mathcal{S}$ to $\mathcal{D}$
9: Generate new $\mathcal{P}$ from $\mathcal{D}$
10: **until** Stopping criteria are satisfied
Output: Best solution and the corresponding function value

In terms of SAEA, it starts by initializing a population $\mathcal{P}$ of N individuals. Each individual in the population is evaluated to form the training dataset $\mathcal{D}$. In the main loop, the surrogate model $\mathcal{M}(\cdot)$ is first trained using $\mathcal{D}$. The algorithm then performs mutation and crossover operations to generate offspring set $\mathcal{P}_o$, following the same procedures as outlined in Algorithm 2 (i.e., the DE algorithm). These operations introduce a series of diversity to the population, enabling the exploration of new regions in the search space. Subsequently, the surrogate model is employed to predict and

prescreen the solutions, identifying the most promising candidates. The prescreening operates similarly to the acquisition function in BO, which rewards solutions either with good prediction and low uncertainty, or with relatively good prediction and large uncertainty.

After prescreening, a small subset $\mathcal{S}$ is selected for real function evaluation by $f(\cdot)$, and this subset is then merged with the training dataset $\mathcal{D}$. When only one new solution is selected for evaluation (i.e., $|\mathcal{S}| = 1$), which aligns with Line 6 of the BO algorithm, SAEA iterates as BO in terms of the cost of expensive function evaluations. Lastly, the algorithm generates a new population from $\mathcal{D}$ to continue into the next loop.

In general, both SAEA and BO optimize the given objective by incorporating machine learning models to reduce and alleviate the need for real function evaluations. Unlike many machine learning-based works in the CAD domain, such as [43, 44, 45, 48, 49], frameworks used in this thesis are free of the need for large-volume datasets for accurate modeling. Instead, the machine learning model is trained solely to indicate promising regions that lead to better solutions (i.e., designs with improved performance metrics). Consequently, in one optimization process, SAEA and BO are able to collect training data, train probabilistic models, and search for the optimal design simultaneously.

As for their difference, SAEAs introduce the specific search engine, i.e. evolutionary operations including mutation, crossover, and selection, into the main loop, whereas BO does not specify any optimizer to solve the model-based sub-problem (2.29). Therefore, BO is more flexible but requires specialized adjustment to adapt to practical problems. SAEA can be directly applied to problems that have been validated by conventional evolutionary algorithms (e.g., DE). Additionally, both frameworks have many variants that claim to possess diverse optimization capabilities.

2.3.1 Handle Prediction Uncertainty

The other important topic to emphasize in the above two algorithms is the method of handling prediction uncertainty, i.e., the acquisition function in BO and the prescreening in SAEA. Here, we introduce three commonly used functions: expected improvement (EI), probability of improvement (PI), and lower confidence bound (LCB) (for minimization problems, while the upper confidence bound (UCB) is used for a

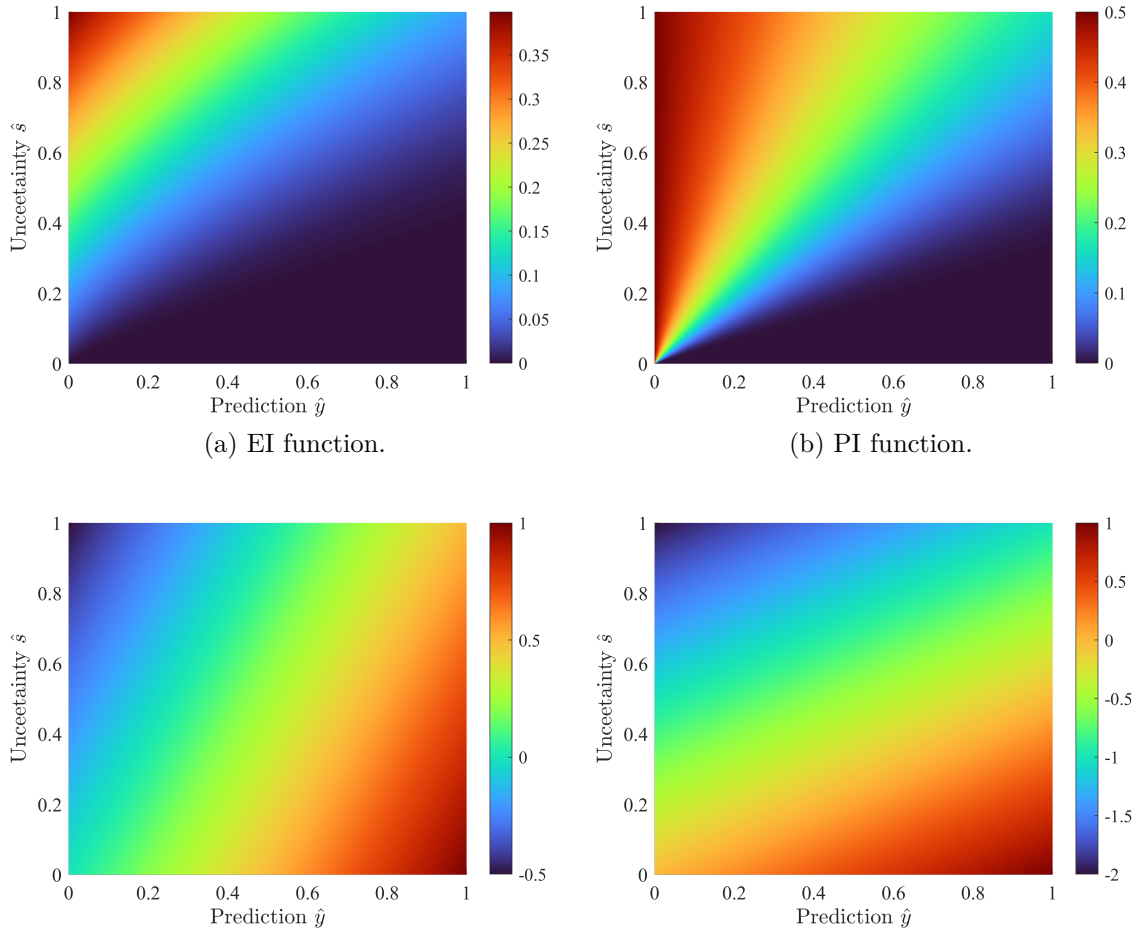
maximization problem). These functions are formulated as follows

$$u_{\mathrm{EI}}(\boldsymbol{x}) = (y_{\min} - \hat{y}(\boldsymbol{x}))\,\Phi\left(\frac{y_{\min} - \hat{y}(\boldsymbol{x})}{\hat{s}(\boldsymbol{x})}\right) + \hat{s}(\boldsymbol{x})\phi\left(\frac{y_{\min} - \hat{y}(\boldsymbol{x})}{\hat{s}(\boldsymbol{x})}\right) \tag{2.30}$$

$$u_{\mathrm{PI}}(\boldsymbol{x}) = \Phi\left(\frac{y_{\min} - \hat{y}(\boldsymbol{x})}{\hat{s}(\boldsymbol{x})}\right) \tag{2.31}$$

$$u_{\mathrm{LCB}} = \hat{y}(\boldsymbol{x}) - \beta\hat{s}(\boldsymbol{x}), \quad \beta \in [0, 3] \tag{2.32}$$

where $\phi(\cdot)$ and $\Phi(\cdot)$ are the probability density function and cumulative distribution function of Gaussian distribution, respectively. $y_{\min}$ denotes the minimum solution of the current iteration (i.e., the current best function value), and $\hat{y}(\boldsymbol{x})$ and $\hat{s}(\boldsymbol{x})$ are the predictive value and standard deviation with regards to the solution $\boldsymbol{x}$. β is the hyperparameter of u_{LCB}, balancing $\hat{y}(\boldsymbol{x})$ and $\hat{s}(\boldsymbol{x})$.



(a) EI function.

(b) PI function.

(c) LCB function when $\beta = 0.5$.

(d) LCB function when $\beta = 2$.

Figure 2.4: Heatmap distribution of three acquisition functions.

The EI, PI, and LCB methods reward predictions with uncertainty in different ways. Figure 2.4 shows the heatmap distribution of three functions, where the y_{min} is set to zero, and $\hat{y}(\boldsymbol{x})$ and $\hat{s}(\boldsymbol{x})$ corresponds to the horizontal and vertical axis, respectively, both varying from zero to one. For EI and PI, higher values indicate better solution quality. EI favors regions with lower predictive values and higher uncertainty, while PI prefers regions with sufficiently good prediction values, regardless of uncertainty. As for LCB, where lower values indicate better solution quality, it offers control over the balance between exploration and exploitation by adjusting the hyperparameter β. When β is relatively small, LCB favors regions with small prediction values. As β increases, greater emphasis is placed on higher uncertainty.

Although comparisons of the above three functions applied in SAEA have shown similar performance in numerical optimization experiments [76], LCB stands out for its flexibility due to the introduction of the controllable hyperparameter β. This hyperparameter is useful for balancing exploration and exploitation in the search process and can be tuned to suit different problems. Therefore, in the following chapters, LCB is often selected as the prescreening method, unless otherwise specified.

2.4 Summary

This chapter introduces the optimization algorithms and machine learning techniques used in this thesis. It begins with an overview of optimization problems, with a particular focus on expensive optimization and black-box optimization. Subsequently, two optimization algorithms are discussed: the local search method, Nelder-Mead simplex, and the global optimization technique, differential evolution. This chapter also delves into various machine learning techniques, including Gaussian process models, deep neural networks, and Bayesian neural networks, explaining their theoretical foundations and practical implementations. Additionally, two optimization frameworks, i.e., BO and SAEA, that integrate these optimization algorithms with machine learning are presented in detail. These frameworks will be adapted to address practical problems in the following chapters. To be specific, Chapter 3 utilizes the SAEA framework with Gaussian process surrogate models and also embeds the Nelder-Mead simplex method. Chapter 4 develops the SAEA framework with alternate DE search operators incorporating Bayesian neural networks. Chapter 5 leverages the SAEA framework and also considers the potent modeling capability of deep neural networks for algorithmic design optimization

Chapter 3

Microwave Filter Design Automation

3.1 Background

Microwave filters are known as the most critical passive components [77] in communication systems, designed to selectively pass or block specific frequencies. Their primary function is to allow signals within a desired frequency band to pass through while attenuating signals outside this band as much as possible. These filters play a vital role in various applications, including satellite communications, radar systems, and mobile networks. Due to the finite resource of the radio spectrum, microwave filters are essential for preventing cross-channel interference and ensuring electromagnetic compatibility [78]. Consequently, the performance of microwave filters is defined by their passband and stopband characteristics, with selectivity being a crucial aspect.

Microwave filters can be classified according to different regards, such as single-mode or multiple-mode filters, single-band or multiple-band filters, and filters with or without cross-couplings. Despite these variations, the main design process generally follows three key steps: topology synthesis, physical dimensioning, and design optimization [79].

Given filter specifications about passband and stopband, topology synthesis [80] begins by approximating the filter's performance requirements into polynomial responses (i.e., the reflection and transmission characteristics, S_{11} and S_{21}). During this step, the filter order, as well as the number and location of transmission zeros (also reflection zeros or

poles hereinafter), are first determined. Ideal values of the lumped-element equivalent circuit or coupling matrix are then calculated within the normalized frequency domain, namely the low-pass prototype. Thanks to extensive research and advances in filter analysis and research on design methodologies, theoretical design can often be achieved through many analytical methods [81, 82]. When analytical synthesis is not feasible, optimization-based synthesis is also straightforward to be employed for obtaining the desired responses[83, 84, 85]. At this point, the filter is theoretically constructed.

The next step, physical dimensioning, translates the theoretical design into practical, real-world implementation [86]. This step starts by aligning real dimensions with denormalized coupling values and subsequently determines the dimension of each resonator or coupling window/gap. Various methods are proposed for physical dimensioning, each with its unique advantages [78, 87, 88, 89]. Most of these methods are fully programmable and can be automated. However, due to the ideal conditions assumed during this process, the full-wave EM response of this initial design often fails to meet the required specifications. As a result, this step is often considered preparatory, leading to a more crucial phase of EM-based design optimization.

Design optimization is the final and most challenging step, essential for achieving stringent specifications required for successful fabrication. During this step, several key design parameters related to resonators and couplings are optimized to approach the desired performance. This task is formidable due to the complex and highly multimodal characteristics of the filter design landscape [90], and it often consumes most time because of the intensive computational costs of 3D EM simulation. To assist designers, experience-guided approaches are often employed, incorporating with off-the-shelf local optimizers which are invoked iteratively and manually by designers. For instance, certain methods proposed in [91, 92, 93] optimize the filter by progressively adding one resonator at a time, and performing an optimization run at each stage. These methods significantly reduce the design space, making the complexity of the design process more manageable within the constraints of design cycles and time-to-market. However, they may not be universally applicable, and their effectiveness can vary depending on the engineer's experience. The outcomes from employing these methods are, therefore, unstable.

Due to the importance of design optimization in filter design, simulation-based design optimization has been attracting much attention since the end of the last century. In recent years, several innovative and successful intelligent 3D design optimization methods have been proposed. For example, space mapping (SM) techniques [94, 95, 96] utilize a low-fidelity model, such as an equivalent circuit, to reduce the required number of computationally expensive EM simulations of high-fidelity. Cognition-driven optimization methods [97, 98] leverage designers' intuition by first optimizing frequency features and then fine-tuning ripple heights. The homotopy method [99] constructs a sequence of intermediate optimization problems that gradually transition from the initial design to the optimal one, proving effective when the initial design is of low quality. Machine learning techniques are employed within some of the above methods to enhance speed and efficiency. Compared to off-the-shelf local optimizers, these methods achieve higher quality solutions more efficiently, with optimization playing a role equally important as designers' expertise. However, these methods were primarily validated only on straightforward filter structures [97, 98, 99], like direct-coupled filters without transmission zeros [100], and their effectiveness for other filter types require further investigation. A more detailed review of their advantages and limitations is provided in the literature review section.

In summary, the design of microwave filters requires a deep understanding of both theoretical principles and practical experience to achieve successful results. Of the three steps outlined, design optimization is the most crucial, as it ensures that the final design meets all required specifications. Therefore, this chapter mainly focuses on the research of design optimization for higher level of filter design automation. It begins with a review of relevant literature to establish the foundational concepts, followed by a summary and clarification of the main problem to be addressed. The proposed methodology is then discussed in detail and demonstrated through several practical design examples, highlighting how this methodology can significantly advance design automation for microwave filters.

3.2 Literature Review

SM is likely the most prevalent attempt in microwave filter CAD since its invention, although it is not specifically designed for microwave filters. To be specific, while the synthesis step produces an ideal design (i.e. an equivalent circuit or coupling matrix) that satisfies all design specifications, a straightforward idea is to build a "mapping"

that correlates the optimal design parameters of circuit or matrix elements with 3D physical structure. This mapping helps guide the design of 3D structures by searching around the optimal region, thereby reducing unnecessary simulation costs [101]. This is the fundamental concept behind the SM method [95]. As the name indicates, SM operates within two design spaces (i.e., the design of lumped elements and 3D structures) aiming to the same design target (i.e., a practical design that meets all performance requirements), with one-to-one correlations between each variable. The mapping captures this correlation, with the equivalent circuit or coupling matrix serving as the coarse model and the 3D EM simulation functioning as the fine model. By iteratively updating this mapping, SM effectively bridges these two models in the near-optimal region, predicting the potential optimal design for the physical structure [96].

Over time, many variants of SM have been proposed and investigated [102, 103, 104, 105]. However, SM has an intrinsic issue regarding design uniqueness, as it assumes that the optimal solution is unique in both design spaces [95]. Algorithms may often struggle to obtain a satisfactory solution when this assumption is not valid. Although several remedies have been proposed [9, 95, 102], their effectiveness is limited due to the local search nature of these methods. For example, a filter tuning method based on SM was proposed in [106], where tunable elements are added to provide circuit-based surrogates at each step of optimization to avoid intensive full-structured EM simulations. However, this method is designed to fine-tune design variables within a small range around the optimal solution and is not capable of designing from scratch, making it less suitable for full automation of filter design. SM is now regarded as an early version of surrogate-assisted optimization (SAO), characterized primarily by its local search capability relying on quasi-Newton methods. In recent decades, more advanced approaches have been developed, specifically targeting filter design problems and surpassing SM in many aspects. Some of these methods are discussed.

In [107], the authors utilize the NM simplex method to optimize ridged waveguide filters characterized by cascaded circuits, where the discontinuities of the filter were represented by generalized scattering matrices. They applied the simplex algorithm iteratively, incorporating additional perturbations to avoid being trapped in local minima. The method is validated by designing two direct-coupled ridged waveguide filters with four and six poles, respectively. However, the authors had to adjust the optimization process to accommodate narrow-band cases, making this method somewhat ad hoc and lacking general design capability. Moreover, ensuring the global optimum becomes challenging when relying solely on the simplex method with perturbations.

Another study in [97] proposed a so-called cognition-driven SM method for the optimization of equal-ripple filters. Intermediate feature metrics, including feature frequency metrics (related to the position of poles) and ripple height metrics (the maximal ripple between two adjacent poles), are extracted from EM simulation to construct two SM models. The optimization process, based on the Trust-region method, unfolds in two stages. First, optimizing the pole positions to fall within the required band, followed by optimizing the ripple height to meet the specified criteria. This method was validated using two direct-coupled filters. However, it may fail if the pole positions in the initial design are not clearly defined, and applying it to filters with designated zeros presents a significant challenge. Additionally, as with the previous method, achieving global optimality is also difficult when using only conventional Trust-region methods.

The concept of feature extraction is further explored in [98] and [108]. In these works, multiple features are extracted from the EM simulation response, including a neural network that maps design variables to a transfer function in a zero-pole format, as well as ripple heights and pole positions. These features mitigate the limitations highlighted in [97], especially when the initial design significantly deviates from the requirements. The transfer function is used to identify pole positions even when they are not explicitly evident. However, this approach still relies on the Trust-region method and has only been verified on direct-coupled filters, making it challenging to apply to general filter design problems.

In most of the aforementioned research, features are extracted from the magnitude of filter responses, whereas in [109] the authors utilize the group delay instead. The coupling matrix of EM responses is first extracted by a global optimizer with its group delay setting as the objective function. Then the SM method with predefined linear mapping guides the search for design parameters. This method was validated using two wideband resonator-coupled bandpass filters and diplexers. However, it is not an end-to-end method; it produces relatively good results but still requires refinement through built-in optimization tools of simulation software, as noted by the authors.

Homotopy optimization is introduced into filter design in [99]. The entire design process is divided into a series of intermediate optimization problems, gradually transitioning from the initial design to the final one. This method greatly reduces the search region and alleviates the challenges faced by local optimizers, potentially leading to convergence on the optimal solution. It was validated with two five-pole waveguide filters.

However, this work does not directly rely on EM simulation; instead, it constructs ANN models for S-parameters of basic sub-blocks (resonators with coupling windows) and then cascades them to characterize filter performance. This method may not be suitable for more complex filter structures. Its general applicability is therefore limited.

The work in [110] presented an innovative global optimization method called SMEAFO, designed for general microwave filter optimization. It combines SAEA with Gaussian local search and was found to surpass existing local optimizers such as SM methods, providing optimal designs that are comparable to the DE algorithm. This algorithm was validated through two examples: a fourth-order direct-coupled waveguide and an eighth-order microstrip filter. Although the algorithm reduces the overall computation time compared to full global optimization, it still requires substantial computational resources, particularly for high-order filters with complex configurations or for large-scale filter design problems.

In [111], a new optimization algorithm, Harris Hawks optimization, inspired by the cooperative behavior of birds, is introduced. It consists of two search phases: the exploration phase, and the exploitation phase. This method was only validated by a four-order dual-mode waveguide filter, where it outperformed many heuristic algorithms, such as DE and PSO. However, applicability to more intricate filters remains to be tested.

With the rise of convolutional neural networks, the authors in [112] leveraged convolutional autoencoder (CAE) as a surrogate model, incorporating particle swarm optimization for microwave filter design. This method follows the typical framework of SAO, with the CAE model being updated online during the optimization process. The CAE effectively represents complete reflection and transmission responses versus frequency, resulting in a highly accurate final design after optimization. However, the initial design in the test case was already very close to the optimized one, with only minor inband violations on ripple, raising questions about its effectiveness in more challenging scenarios.

In summary, the above literature can be grouped into three categories based on the degree of human participation. Methods proposed in [96, 101, 106] can be classified as *supervised* design optimization, where the designer's tuning procedure and experience play a major role in determining the final design success. These approaches are currently the standard routine approach in industrial design practices. Methods proposed in [97, 98, 99, 108, 109] can be considered as the *semi-supervised*, which still require designer

interaction for peripheral tasks and to help the optimizer escape local optima. In these methods, the success of the design is determined by both the practitioner and the optimization algorithm. Methods proposed in [110, 112] have the potential to become real *unsupervised* design optimization by extensive validation for their effectiveness and efficiency across more general design cases. However, due to the highly multimodal characteristics of the filter design landscape, as mentioned earlier, this is often not guaranteed naturally.

3.3 Problem Description

There are at least two characteristics that must be fulfilled for an *unsupervised* design optimization method: 1) it is end-to-end to satisfy stringent design specifications without extra steps, making designer interactions unnecessary during the optimization process. In other words, the process is merely completed simply by launching the program, with no need for further consideration. 2) It must be applicable to most design cases and is not restricted to specific filter types or structures.

Benefits of this methodology are evident: 1) It will significantly reduces the design time (and thus the cost) required by engineers; 2) it can be applied to most engineers with less design experience, while still ensuring successful outcomes.

Figure 3.1 summarizes the input and output that are considered in the proposed unsupervised design optimization. It takes design specifications, the ideal response, the initial design structure, and specified design parameters as inputs, and outputs the optimal design structure discovered by algorithms. All inputs are considered the essential information and knowledge needed for designing a microwave filter. In addition, the ideal response is derived from topology, which implicitly indicates the ideal positions of poles and zeros. For a typical bandpass filter, design specifications might include center frequency, filter bandwidth, passband reflection level, and stopband transmission level. To ensure that the algorithm is applicable to a wide range of filter design problems, no additional specific design knowledge or techniques are leveraged. As discussed, this is critical for achieving effective *unsupervised* filter optimization.

To achieve this goal, the following elements are considered essential:

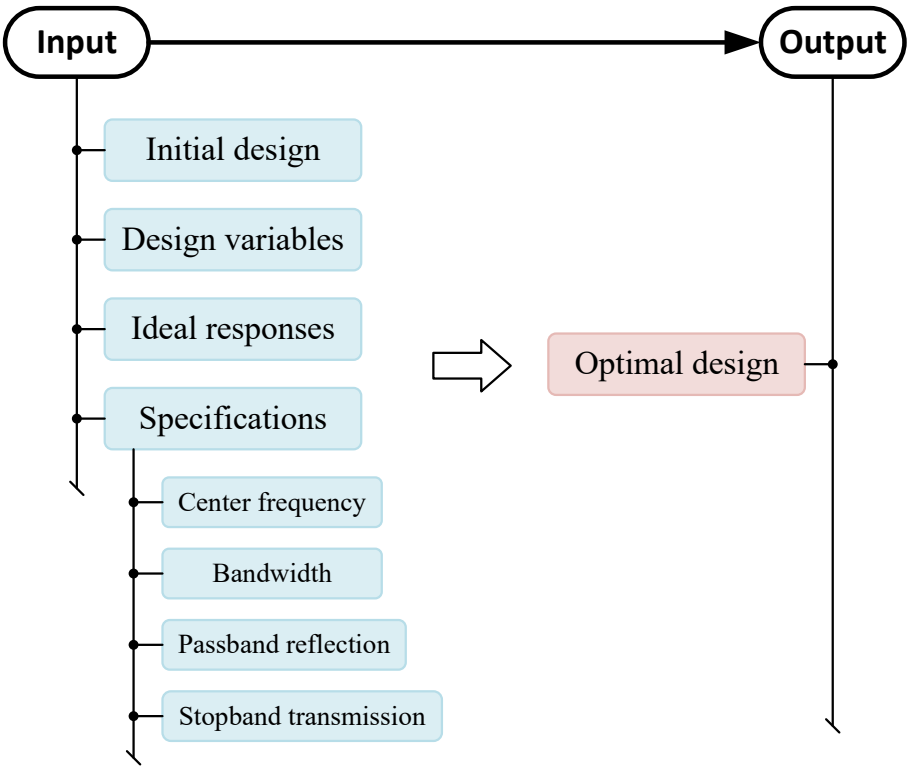


Figure 3.1: Illustration of input and output for filter design optimization problem.

- A framework or methodology that effectively integrates filter design knowledge (as outlined in Figure 3.1, the inputs) with optimization algorithms, while maintaining reasonable versatility across a diverse type of filters.
- Proper objective functions that simplify the filter design landscape. Besides objective functions based on the magnitude of S-parameters (e.g., minimizing the maximum magnitude of $|S_{11}|$ in dB within a required band, denoted by $\max(|S_{11}|)$), which are straightforward and very widely used, several promising methods have also been proposed [98, 99, 109]. However, these works are far from mature, and it remains unclear whether these functions are still valid for an end-to-end framework with broader applicability. We compare some of these functions in the experiment section of this chapter.
- A global optimization algorithm bespoke for filter design problems. Compared to local optimization, global optimization ensures that the optimal design is likely to be found, and improves the success rate of outcomes. However, due to the no-free-lunch theorem [113], conventional global optimization is often slow and inappropriate to be applied directly. This necessitates the development of a novel algorithm that is specialized for filter design problems.

These three elements are deemed indispensable for achieving automated filter design optimization, yet no research in the literature fully addresses all of them. For instance, in [110], a promising endeavor called SMEAFO is proposed for general microwave filter optimization, applying global optimization based on SAEA. However, it lacks sufficient consideration of the three elements mentioned above, particularly overlooking systematic design knowledge, and thus only targets small-scale design problems (less than 6 design variables) without specification on stopband. Nevertheless, SMEAFO provides valuable insight into the intrinsic challenge of filter design problems and lays a pivotal foundation for further research. In the next section, the proposed methodology is elaborate, incorporating some operators inherited from SMEAFO.

3.4 Proposed Methodology

The pipeline of the proposed methodology is illustrated in Figure 3.2. It consists of three main blocks in addition to the input and output stages. Design knowledge utilized in each block is displayed above the corresponding block, while the objective function and optimization engine are listed below, respectively. The workflow can be summarized as follows. The initial design obtained from physical dimensioning is first accepted as the input of the entire design process. Based on this initial solution, a systematic sampling method is performed to sample a set of designs (the initial design dataset) around it, with resonator theory being employed to determine the appropriate perturbation that ensures effective sampling. Subsequently, the optimization stage unfolds in two phases, each involving different design knowledge, objective functions, and search engines. Phase I optimization aims to quickly converge on a solution that captures the general shape (i.e., locate the positions of zeros and poles calculated by the ideal response) of the desired response, while Phase II optimization focuses on finding the optimal design that satisfies all design specifications. Once the entire process is complete, the best solution in the design set is output. Note that, although the initial design obtained by any physical dimensioning methods is acceptable, the recommended programmable method is described in [114], which has been found efficient and effective by incorporating with the proposed pipeline in this chapter.

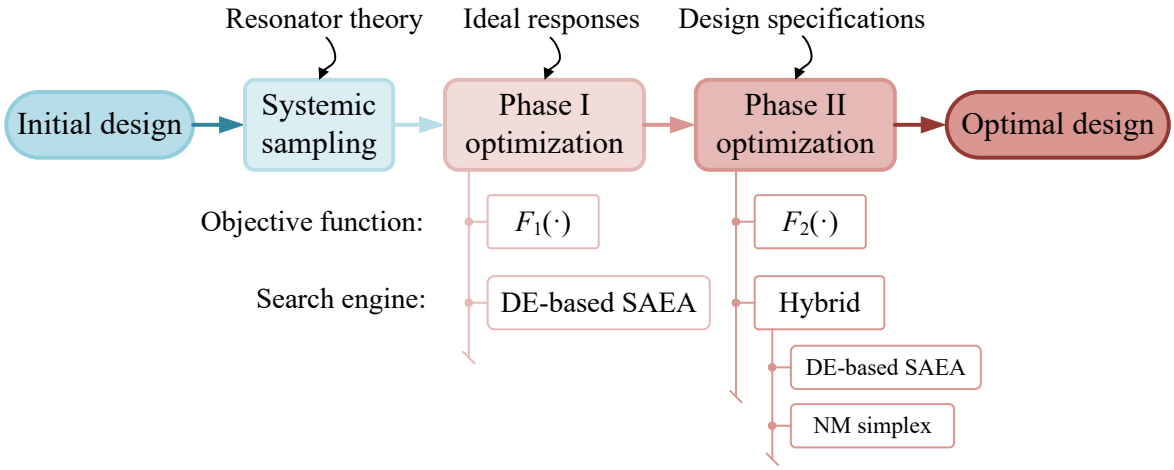


Figure 3.2: Illustration of the pipeline of the proposed methodology.

Two questions need to be clarified regarding this methodology: First, why is the systematic sampling method emphasized and considered indispensable? Second, why is it necessary to divide the optimization process into two phases, each utilizing different objective functions and search engines? We briefly address these questions and then provide a detailed description of associated techniques in the following subsections.

The initial design of the microwave filter is of great importance in filter design optimization due to its complicated landscape, i.e. the narrow optimal region and numerous local optima. It provides critical information about the potential search region, though its performance is often poor. However, global optimization, particularly DE, is generally considered free of initial solutions, as it uses the difference between individuals in the population to maintain search capacity. Thus, pervasive sampling across the entire design space can undermine the benefits of a good initial design and hinder convergence. Conversely, undersampling solely around the initial design may weaken the algorithm's search capacity. To resolve this conflict and strike a balance—leveraging the information from the initial design while integrating it effectively into the downstream algorithm—a systematic sampling method is crucial and essential. Therefore, an appropriate sampling method forms the foundation for the subsequent optimization process and cannot be overemphasized.

Regarding the use of different objective functions, despite the advantages of a good initial design and systematic sampling method, their performance is often far from requirements. This makes some objective functions difficult to assess the quality of designs at the early stages of optimization, especially when using the straightforward one—minimizing $\max(|S_{11}|)$. This issue can significantly slow down convergence or

even prevent it altogether. For instance, Figure 3.3 shows two $|S_{11}|$ responses of a filter that needs to operate from 4.9 to 5.1 GHz. While they differ in quality from a designer's perspective, they have very similar objective value regarding $\max(|S_{11}|)$: -3.36 dB versus -3.15 dB. It's inapplicability is thus demonstrated clearly.

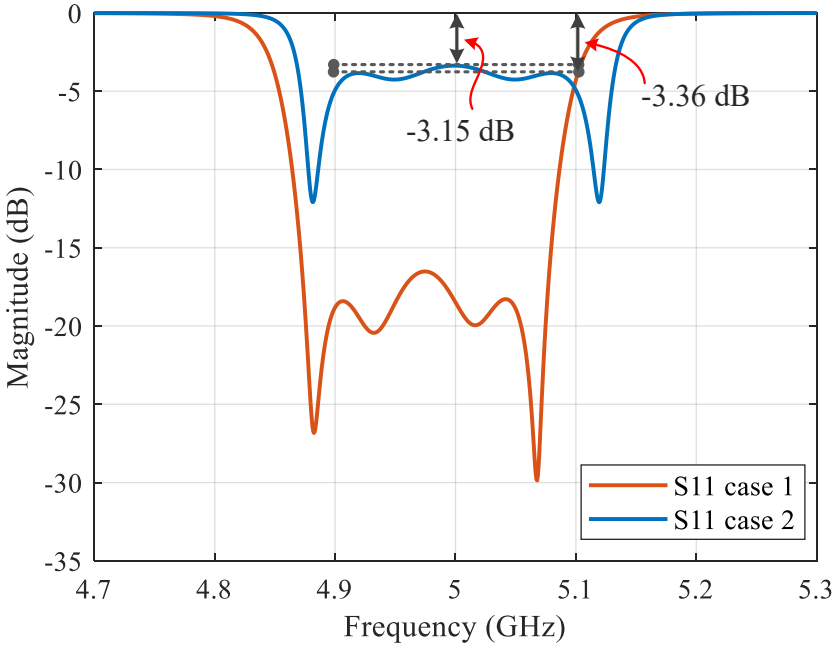


Figure 3.3: Illustration of the issue of $\max(|S_{11}|)$ objective function in two typical cases.

In general, an appropriate objective function should: 1) align with the design specifications, 2) effectively discriminate between different designs, and 3) Smooth the design landscape near the global optimum by preventing penalties for values that exhibit drastic changes. Clearly, minimizing $\max(|S_{11}|)$ in dB only meets the first criterion. Therefore, applying different objective functions at various stages is a practical approach to meet these requirements throughout entire optimization process and to ensure the success of an unsupervised design methodology.

As for the search engine, the methodology primarily relies on SAEA framework, with the NM simplex method being employed in Phase II to accelerate convergence. To ensure the success rate of the proposed algorithm, exploration ability must be maintained to avoid local optima, particularly at the beginning of optimization, as in Phase I. In Phase II, pilot experiment indicates that hybridizing SAEA with a local search engine benefits rapid convergence, even though the exploration ability is somewhat compromised. This trade-off is acceptable given the payoff—if the current best design is improved through local search, the entire population benefits in subsequent iterations. This approach makes the design optimization process both efficient and effective.

3.4.1 Systematic Sampling Method

The aim of sampling is to provide a basic understanding of the characteristics of the design landscape and to form the initial population for optimization algorithms. Several conventional sampling methods are widely used in SAEAs, such as full-factor sampling [115], Monte Carlo sampling [116], and Latin hypercube sampling [117]. However, these methods mainly focus on maintaining the uniformity of samples in the design space, thus losing the advantage of the initial design.

To preserve the benefits of the initial design while still providing sufficient diversity across the design space, a feasible approach is to perturb each element of the initial design by adding Gaussian-distributed random numbers with zero mean and specified variances. This is formulated as follows:

$$x_i = x_i + \mathcal{N}(0, \sigma^2),\ i \in \mathcal{I} \tag{3.1}$$

The value of σ is crucial in this method. It determines how much information from the initial design is utilized. If a large random number is added to the initial design parameters, the pattern of the initial design may be overwritten, rendering it less useful to the global optimizer. In contrast, adding a small random number retains patterns of the initial design, but may cluster the initial samples within a narrow region, greatly preventing the search from escaping local optima.

In this context, σ measures the dimensional perturbation for initial geometries. Therefore, according to microwave resonator theory [118], for a resonator with a physical length L, L is proportional to the guided wavelength λ_g at the resonant frequency f. Therefore, the perturbation ΔL of the physical length and the corresponding frequency shift Δf are related as follows

$$\Delta L \propto \frac{\Delta f}{f}\lambda_g \tag{3.2}$$

Thus, L varies with FBW $\times\ \lambda_g$, where FBW can be cast as the fractional bandwidth of a filter. This relationship forms the solid foundation of the parameter-perturbation method and is broadly applicable for different filter types. Additionally, design variables are divided into two categories: resonance-related and coupling-related, covering most cases of filter design variables. Resonance-related variables primarily control

the center frequency and bandwidth, directly influencing the frequency characteristics. Coupling-related variables, on the other hand, mainly control ripple heights. According to microwave theory, these variables exhibit different sensitivities in terms of the filter response. Therefore, σ is assigned a different value for each category.

Based on pilot experiments, for resonating-related design variables, let $\sigma_f = 0.25 \times FBW \times \lambda_g$ and for coupling-related variables, let $\sigma_c = FBW \times \lambda_g$, where λ_g is the guided wavelength of the central frequency. This empirical rule is also applicable to other global optimizers for filters.

In summary, the pseudo-code of the proposed systematic sampling method is shown in Algorithm 5, where R represents the output response (i.e., the S_{11} and S_{21}) from the EM simulation, and $\texttt{randn}(0, \cdot)$ denotes the Gaussian random number with zero mean and a given variance. Once the algorithm is complete, the sampling dataset $\mathcal{D}$ is output.

Algorithm 5 Filter Systemic Sampling Method (`FilterSampling`$(\cdot)$)

Input: Initial design $\boldsymbol{x}_{\text{ini}}$, the number of samples N, simulation program `Sim`()

$R \leftarrow \texttt{Sim}(\boldsymbol{x}_{\text{ini}})$
$\mathcal{D} \leftarrow \{(\boldsymbol{x}_{\text{ini}}, R)\}$
$n \leftarrow 1$
repeat
  $\boldsymbol{x} \leftarrow \boldsymbol{x}_{\text{ini}}$
  for each element x_i in $\boldsymbol{x}$ **do**
    if x_i is coupling-related **then**
      $x_i \leftarrow x_i + \texttt{randn}(0, \sigma_c^2)$
    end if
    if x_i is resonance-related **then**
      $x_i \leftarrow x_i + \texttt{randn}(0, \sigma_f^2)$
    end if
  end for
  $R \leftarrow \texttt{Sim}(\boldsymbol{x})$
  $\mathcal{D} \leftarrow \{(x, R)\} \cup \mathcal{D}$
  $n \leftarrow n + 1$
until $n > N - 1$

Output: Initial sampling dataset $\mathcal{D}$

3.4.2 Phase I Optimization

3.4.2.1 Objective Function: F_1

In the initial phase, many points around the initial design may exhibit poor responses. Thus, the goal of Phase I is to quickly obtain a general shape of the desired response, i.e., determine the positions of zeros and poles. Therefore, the proposed objective function is based on the response at the ideal positions of zeros, poles, and edge (denoted as ZPE function) [83] with an added term Z to restrict the bandwidth, formulated as follows:

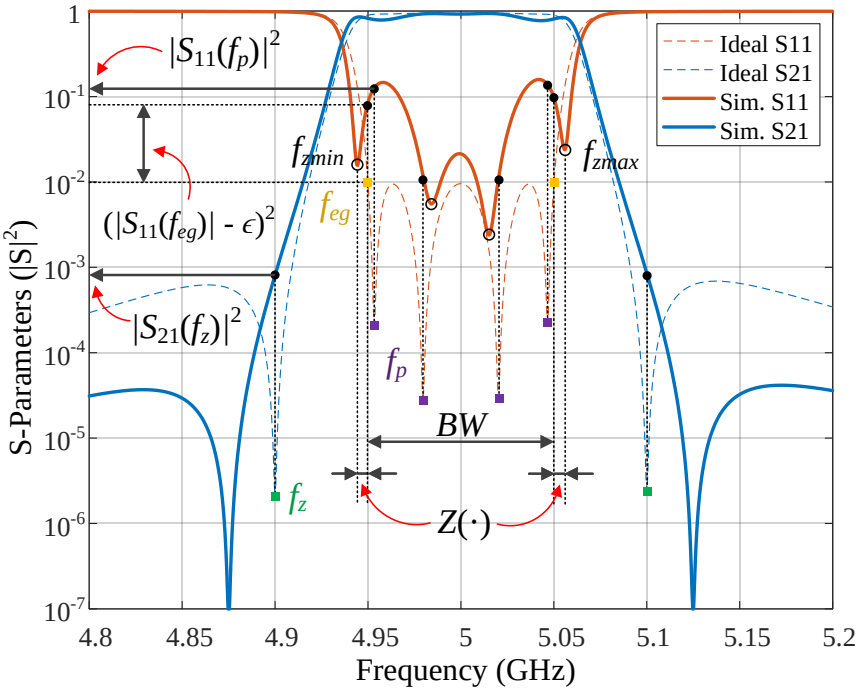


Figure 3.4: Illustration of key features considered in F_1

$$F_1 = \sum_i |S_{21}(f_{z_i})|^2 + \sum_j |S_{11}(f_{p_j})|^2 + \sum_k (|S_{11}(f_{eg_k})| - \epsilon_{\text{eg}})^2 + Z(BW) \tag{3.3}$$

where ϵ_{eg} is the magnitude of S_{11} at the edge of the passband under the ideal condition, and all S-parameters are real values. BW are the bandwidth requirement of the filter. By considering the magnitude of S-parameters at the ideal frequencies of zeros, poles, and edges (denoted by f_{z_i}, f_{p_j} and f_{eg_k}), the desired response shape can be efficiently formed. Key features of this proposed objective function are depicted in Figure 3.4.

This ZPE objective function was originally developed for optimization-based filter CM synthesis and has shown high efficiency in [83]. Some pilot experiments show that it is much more efficient at obtaining a coarse shape of the desired response compared to minimizing $\max(|S_{11}|)$ for the proposed design methodology.

The added term $Z(BW)$ is defined as

$$Z(BW) = (f_{z_{\max}}(S_{11}) - f_{z_{\min}}(S_{11}) - \epsilon_{\mathrm{BW}} - BW)/BW \tag{3.4}$$

where $f_{z_{\max}}(S_{11})$ and $f_{z_{\min}}(S_{11})$ are the maximum and minimum frequency from extracted reflection zeros, identified through vector fitting [119, 120, 121]. ϵ_{BW} represents the small difference between the bandwidth estimated from the extracted reflection zeros and the required bandwidth, calculated theoretically. Vector fitting can accurately identify reflection zeros even for designs with poor responses, as validated by existing research [98]. The division by BW serves a normalization purpose. The role of the Z term is to penalize designs with a typical incorrect response shape, where some of the reflection zeros fall outside the passband, even though ZPE function may meet the passband specification. Additionally, no weighting is required, as all terms are comparable in scale due to normalization and are equally important. Specifically, for an ideal response, all terms in F_1 should be zero.

Nonetheless, this objective function cannot be used throughout two phases. This is because there is often a deviation between the ideal response and the real response obtained from physical design. In most cases, even when the filter is fully optimized using F_1, the specifications are often not met, in other words, F_1 does not always align with the design specifications while the general response shape has been formed. Consequently, a new objective function is introduced in Phase II, as described in the next section.

3.4.2.2 Optimization Algorithm: DE-based SAEA

With the proposed objective function F_1, the pseudo-code for Phase I optimization is shown in Algorithm 6, where the main framework is inherited from [110]. Compared to the general framework of SAEA in Algorithm 4, several modifications are made to adapt to filter design problems. First, the top N designs, ranked by F_1, are selected at the beginning of the main loop, ensuring that the population is consistently com-

posed of elite solutions in each iteration. Second, mutation and crossover operations are performed following the same approach in DE (Algorithm 2), where the DE/current-to-best/1 mutation strategy is used to balance the exploration and exploitation. As discussed in Section 2.1.3, this strategy promotes the spread of good patterns from the best solution while maintaining sufficient diversity through the current solution.

Algorithm 6 Phase I Filter Design Optimization

Input: Initial sampling dataset $\mathcal{D}$, objective function F_1, population size N, simulation program $\texttt{Sim}(\cdot)$

1: **repeat**
2: Sort all designs in $\mathcal{D}$ in ascending order by F_1
3: Select top N designs to form the population $\mathcal{P}$
4: Perform mutation and crossover operators on $\mathcal{P}$ to form $\mathcal{P}_o$
5: Train GP models $\{\mathcal{M}(\cdot)\}$ for each solution in $\mathcal{P}_o$
6: Predict and prescreen solutions in $\mathcal{P}_o$ by GP models
7: Select the best $\boldsymbol{x}_o \in \mathcal{P}_o$ with the minimium prescreening value
8: Simulate $\boldsymbol{x}_o$ by $\texttt{Sim}(\cdot)$ and add $\boldsymbol{x}_o$ with its simulation result to $\mathcal{D}$
9: **until** Average improvement of F_1 less than 2%

Output: Current best design $\boldsymbol{x}_{\text{best}}$, simulated dataset $\mathcal{D}$

Additionally, GP models are employed as surrogates. To ensure high model quality, models are constructed for each solution in $\mathcal{P}_o$ and for each term of F_1 in one iteration, resulting in a total of $4N$ local GP models per loop. Training separate models for each term of the objective function, as well as for each solution has been shown to perform better than training a single model for the summation. Regarding the training dataset, it is selected by calculating the Euclidean distance between the current solution and individuals from $\mathcal{D}$. The nearest N individuals are chosen to train a local GP model.

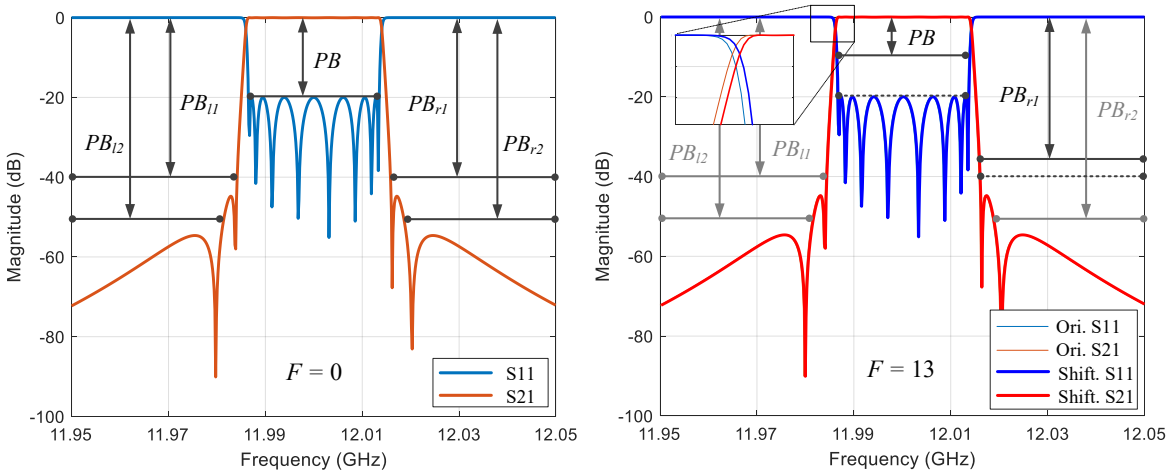
For the prediction and prescreening operation outlined in Line 7, the LCB method is employed, with coefficient β being 2. Afterward, the best solution from $\mathcal{P}_o$ in prescreening is simulated, meaning that EM simulation is carried out once per iteration. Therefore, the efficiency of the algorithm can be compared based on the number of iterations or the number of simulation calls.

Phase I optimization stops when the average improvement of F_1 is less than 2% over consecutive 100 iterations. This 2% is empirical and not particularly sensitive. Once Phase I concludes, Phase II optimization is launched, with an objective function focused on exactly satisfying the specifications and smoothing the design landscape at the same time. This helps to reduce the burden on the global optimizer and accelerates the search process described in the next section.

3.4.3 Phase II Optimization

3.4.3.1 Objective Function: F_2

In Phase II optimization, the objective function is defined to find the optimal design that meets all specifications. A straightforward idea is to extend the $\max(|S_{11}|)$ to $\max(|S_{11}|) + \max(|S_{21}|)$, in which passband performance is penalized by $\max(|S_{11}|)$, and stopband performance is penalized by $\max(|S_{21}|)$. When the function value approaches zero, all design specifications should be fully met. However, as discussed earlier, $\max(|S_{11}|)$ is not ideal for use at the beginning of optimization due to its low discriminative ability, and it is also not suitable for Phase II, despite aligning with required specifications.



(a) Optimal responses (b) Shifted responses

Figure 3.5: Comparison of $\max(|S_{11}|) + \max(|S_{21}|)$ values for a practical example.

$$
\begin{aligned}
F =& \max(PB - (-20), 0) + \max(PB_{l_1} - (-40), 0) + \\
& \max(PB_{l_2} - (-40), 0) + \max(PB_{r_1} - (-50), 0) + \\
& \max(PB_{r_2} - (-50), 0), \\
PB =& \max(|S_{11}|), \text{ in dB from 11.9865 to 12.0135 GHz} \\
PB_{l_1} =& \max(|S_{21}|), \text{ in dB from 11.9500 to 11.9840 GHz} \\
PB_{l_2} =& \max(|S_{21}|), \text{ in dB from 12.0160 to 12.0500 GHz} \\
PB_{r_1} =& \max(|S_{21}|), \text{ in dB from 11.9500 to 11.9810 GHz} \\
PB_{r_2} =& \max(|S_{21}|), \text{ in dB from 12.0190 to 12.0500 GHz}
\end{aligned}
\tag{3.5}
$$

To elaborate its issue, consider a practical example shown in Figure 3.5, which illustrates the ideal response of an eighth-order filter and a slightly shifted response (where all S-parameters shift 0.00025GHz to higher frequency). This results in a dramatic change in the objective function from 0 to 13, as formulated in Equation (3.5), which is very counter-intuitive. Unarguably, this objective function will lead to a rugged landscape [90, 110]. The reason behind this issue is that as the order of the filter increases and its edge response becomes steeper, small perturbations in the design variables can cause drastic changes in the objective function, rendering this value meaningless. Therefore, proposing a new objective in Phase II becomes inevitable.

The proposed objective function for Phase II is denoted by F_2 as shown in Figure 3.6, which includes three terms: the average inband ripple $R(\cdot)$, stopband edge frequency $E(\cdot)$, and bandwidth restriction $Z(\cdot)$. All of them are to be minimized.

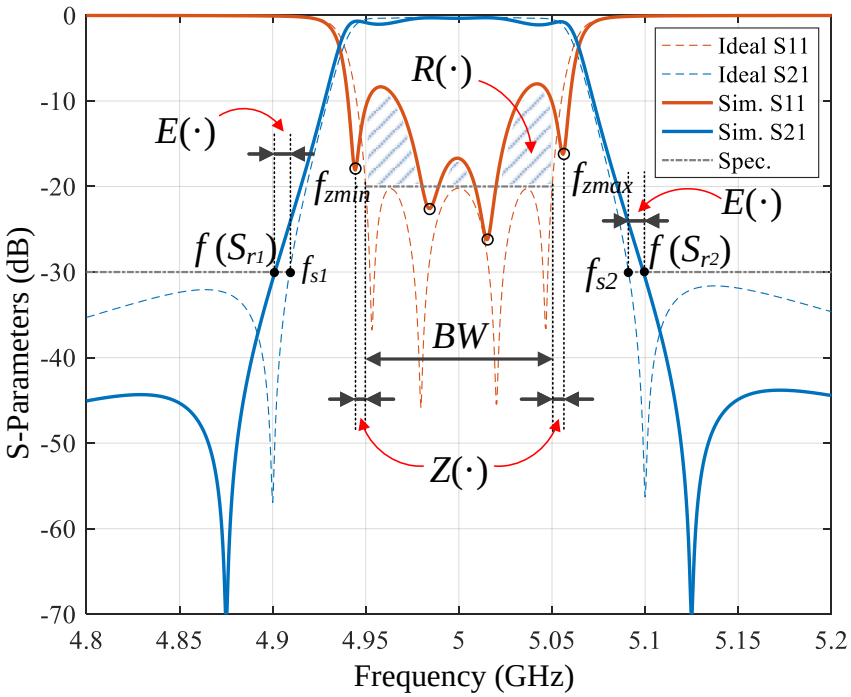


Figure 3.6: Illustration of key features considered in F_2

The inband ripple term ($R(\cdot)$ in Figure 3.6) is defined as:

$$R(BW, S_r) = \frac{1}{S_r BW} \int_{BW} \max(|S_{11}(f)| - S_r, 0) \mathrm{d}f \tag{3.6}$$

where BW is the bandwidth, S_r is the specification (e.g., -20 dB for inband $|S_{11}|$). $\frac{1}{S_r BW}$ is a normalization factor. The use of integration calculates the average violation of the S_{11} specification, effectively smoothing the design landscape and preventing the issue from using $\max(|S_{11}|)$ as discussed above. Pilot experiment shows that with the same optimizer, not only is the success rate improved, but the convergence speed is also much faster compared to using $\max(|S_{11}|)$.

The stopband edge frequency term ($E(\cdot)$ in Figure 3.6) is defined as:

$$E(\boldsymbol{f_s}, \boldsymbol{S_r}) = \sum_i \max(f(S_{r_i}) - f_{s_i}, 0)/BW \tag{3.7}$$

where $f(\boldsymbol{S_r})$ is the first frequency where $|S_{21}|$ meets the given stopband specification $\boldsymbol{S_r}$ (e.g., $|S_{21}|$ reaches -30 dB at the frequency $f(-30\text{dB})$), and $\boldsymbol{f_s}$ is the frequency specification (e.g., $|S_{21}|$ should be under -30 dB at $\boldsymbol{f_s}$ and below). The index i indicates the number of stopband specifications. The division by BW serves as normalization. Note that even for some design cases without stopband specifications, this term can be added using a reasonable estimation of $\boldsymbol{f_s}$. Pilot experiment shows that this improves optimization speed in collaboration with the previous term.

The passband reflection zero term (i.e., $Z(\cdot)$ in Figure 3.6) $Z(BW)$, is the same as that in F_1. To sum up, the final objective function for Phase II is

$$F_2 = R(BW, \boldsymbol{S_r}) + E(\boldsymbol{f_s}, \boldsymbol{S_r}) + Z(BW) \tag{3.8}$$

Note that no weighting is needed, as the terms are normalized. In comparison with the objective functions formulated in Equation (3.5), for the same responses shown in Figure 3.5, F_2 varies only from 0 to 0.674, which aligns with human intuition and results in a smoother design landscape.

3.4.3.2 Optimization Algorithm: Hybrid DE-based SAEA with NM Simplex

The pseudo-code of the Phase II optimization using F_2 is shown in Algorithm 7. Lines 2 to 8 follow the same processes performed in Phase I. From lines 9 to 12, when the current best solution is not updated for 50 consecutive iterations, the local optimizer (i.e., `NMSimplex`$(\cdot)$) is triggered. Once the local optimizer reaches a local optimum

Algorithm 7 Phase II Filter Design Optimization (Hybrid algorithm)

Input: Design dataset $\mathcal{D}$, objective function F_2, population size N, simulation program $\texttt{Sim}(\cdot)$

1: **repeat**
2: Sort all designs in $\mathcal{D}$ in ascending order by F_2
3: Select top N designs to form the population $\mathcal{P}$
4: Perform mutation and crossover operators on $\mathcal{P}$ to form $\mathcal{P}_o$
5: Train GP models $\{\mathcal{M}(\cdot)\}$ for each solution in $\mathcal{P}_o$
6: Predict and prescreen solutions in $\mathcal{P}_o$ by GP models
7: Select the best $\boldsymbol{x}_o \in \mathcal{P}_o$ with the minimium prescreening value
8: Simulate $\boldsymbol{x}_o$ and add $\boldsymbol{x}_o$ with simulation result to $\mathcal{D}$
9: **if** $\boldsymbol{x}_{\text{best}}$ remains unchanged for 50 iterations **then**
10: Perform local search $\texttt{NMSimplex}(\boldsymbol{x}_{\text{best}}, F_2)$
11: Select and add solutions visited by local search to $\mathcal{D}$
12: **end if**
13: **until** Stopping criteria are satisfied

Output: Best solution $\boldsymbol{x}_{\text{best}}$ and the corresponding function value

or the maximum number of iterations, it halts, and the design dataset $\mathcal{D}$ is updated by merging selected solutions visited by the local optimizer. Therefore, compared to Algorithm 6, this algorithm hybridizes DE-based SAEA with NM simplex, where the two optimizers complement each other by combining their strengths. Hereinafter, the algorithm proposed for Phase II in this chapter is also called the *hybrid* optimization algorithm.

As introduced in Section 2.1.2, NM simplex is a derivative-free search method that is well-suited for rugged, highly multimodal landscape [60], which is a common characteristic of filter design landscapes. In the local optimization process, no surrogate model is used. This is because local search requires a highly accurate surrogate model, and building one with a limited number of EM simulations is often challenging in filter design. An inaccurate surrogate model could mislead the local search. Pilot experiments with real-world filters show that without using a surrogate model converges even faster than using a GP model in NM simplex optimization.

Furthermore, solutions visited by the local optimizer must be carefully selected for inclusion in the database, considering both performance and diversity. This is essential to avoid SAEAs being trapped in local optima, especially when the current best N solutions in the database lack diversity. The scoring strategy from [122] is used to rank and select solutions visited by NM simplex, which are then added to the design dataset as outlined in Line 11.

3.5 Experiment Results

The performance of the proposed methodology is validated through two real-world examples, including an eighth-order dual-band waveguide filter with four transmission zeros [123] and a sixth-order waveguide filter with two transmission zeros [124]. The initial design for both filters is obtained using the method presented in [114]. As demonstrated in the subsequent experiments, several widely used filter optimization methods fail to achieve unsupervised design when starting from the given initial designs. All the experiments are conducted on a workstation with Intel 3.2 GHz Core (TM) i7 CPU and 8 GB RAM, running the Windows operating system. The simulations are performed using CST Microwave Studio, with wall clock time used to measure time consumption. No parallel computing is employed.

To highlight the advantages of the proposed new objective functions, the same optimization algorithm used in Phase II is applied with the same initial design to compare several reference objective functions. These include: (a) the sole use of F_1 without bandwidth constraint (i.e., ZPE objective function in [83]), (b) the sole use of F_2, (c) the objective function based on extracted CM from group delay [109] (referred to as CM-difference method in the following), (d) the cognition-driven multi-feature objective function [98], and (e) the commonly used S-parameters based objective function (e.g., minimizing $\max(|S_{11}|) + \max(|S_{21}|)$, simplified as $\max(|S_{11}|)$ in the following).

Each objective function is carried out with five independent runs, and the results are compared statistically. Due to the computational cost of EM simulations, more runs are not affordable. In all the comparisons, a run is considered successful if the optimal design achieves less than 0.1 overall specification violation (i.e., summed together) within 2500 EM simulation budget. Note that this limit is only for comparison purposes; the proposed methodology required significantly fewer EM simulations than this budget.

To demonstrate the benefits of the hybrid optimization algorithm proposed in Phase II, a comparison is conducted between two scenarios: the one where the NM simplex is involved (hybrid) and the one where it is not (nonhybrid). This comparison highlights the impact of integrating local search within SAEA for filter design optimization. Note that the nonhybrid version behaves the same as the algorithm in Phase I. In other words, the comparison is carried out between Algorithm 7 and 6, with the same objective function (i.e., F_2).

Due to the stochastic nature of algorithms, random seeds influence the number of EM simulations required to satisfy the specifications. To minimize this effect and focus on comparing different search mechanisms, the experiments are divided into five groups. In each group, initial populations are randomly generated by the systematic sampling method, and the same initial population and random seed are applied for five independent runs with the hybrid and nonhybrid algorithms. The convergence speed is compared statistically. All five groups yield the same conclusion. Hence, the results from one typical initial population are displayed in the following subsections.

3.5.1 Example 1: X-band Dual-band Filter

The first example is an X-band symmetric eighth-order dual-band filter with four transmission zeros, as shown in Figure 3.7. This filter is designed to operate at the center frequency of 10 GHz with two passbands symmetrically located at 9.35-9.70 GHz and 10.30-10.65 GHz. Of the eight cavity resonators, four are independent and the corresponding resonators share the same dimensions. The four resonators (i.e., resonators 1-4 or 5-8) form a cascaded quadruplet which generates two explicit transmission zeros at 9.88 GHz and 10.12 GHz, respectively. Due to the symmetric structure, a total of four transmission zeros are created but overlapped at two frequencies, resulting in the improvement of stopband rejection between two passbands. This filter has 10 design variables, of which $[L1, L2, L4]$ $(L2 = L3)$ target the resonant frequency, and $[W12, W34, W14, W45, We, H23, L23]$ control the coupling between resonators. The filter is modeled in CST Microwave Studio with around 12000 mesh elements, and each EM simulation takes 1 and 1.5 minutes to complete.

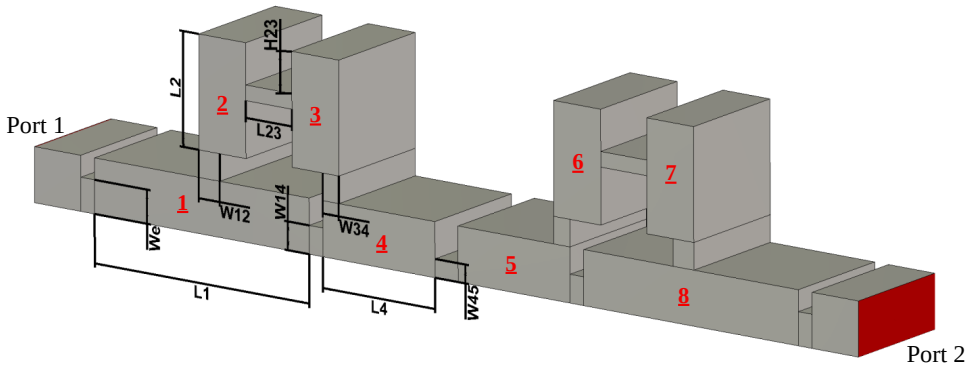


Figure 3.7: The structure of the X-band filter

Table 3.1: Design specifications for example 1

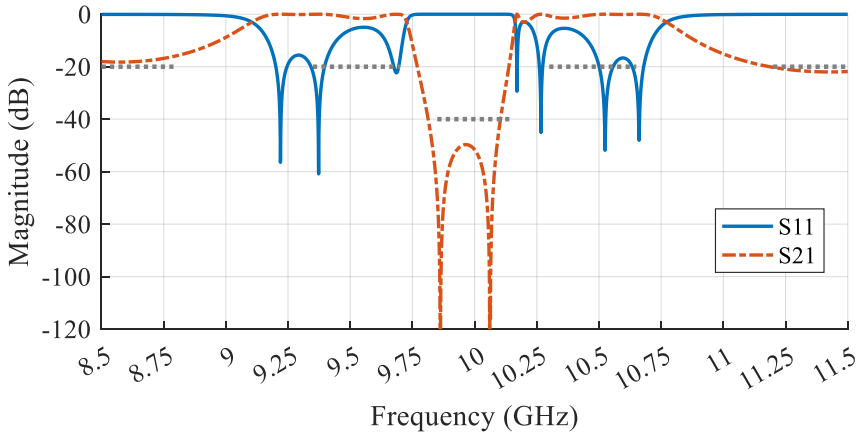
Notation	**Item**	**Frequency Range** (GHz)	**Specification** (dB)
PB_1	Passband 1 Reflection Coefficient (S_{11})	9.35 - 9.70	-20
PB_2	Passband 2 Reflection Coefficient (S_{11})	10.30 - 10.65	-20
SB	Stopband Transmission Coefficient (S_{21})	9.85 - 10.15	-40
SB_l	Stopband Left Edge Transmission Coefficient (S_{21})	≤ 8.8	-20
SB_r	Stopband Right Edge Transmission Coefficient (S_{21})	≥ 11.2	-20

Table 3.2: The initial design and a typical optimized design (all sizes in mm) (example 1)

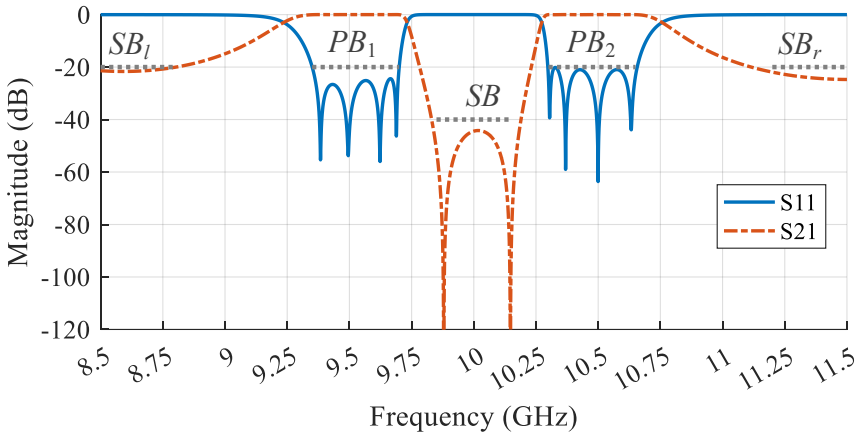
Variable Name	$W12$	$W34$	$W14$	$W45$	We
Initial Value	4.504	3.464	5.210	3.417	6.186
Optimized Value	4.234	3.489	4.192	2.713	6.894
Variable Name	$H23$	$L23$	$L1$	$L2$	$L4$
Initial Value	7.645	9.940	46.374	20.983	24.193
Optimized Value	6.766	8.962	46.035	20.673	23.359

The design specifications are listed in Table 3.1, with notations marked in Figure 3.8(b). Note that the central stopband is formed by four transmission zeros, with two located at 9.88 and the other two at 10.12 GHz.

In all five runs, the optimal design obtained after the two-phase optimization process successfully meets all specifications. The initial design and a typical optimal design are shown in Table 3.2, with the response shown in Figure 3.8. Across the five runs, an average of 678 EM simulations were executed, taking 13 hours. This demonstrates that the time consumption is manageable compared to the manual design process from industry, especially considering an *unsupervised* design process without human intervention.



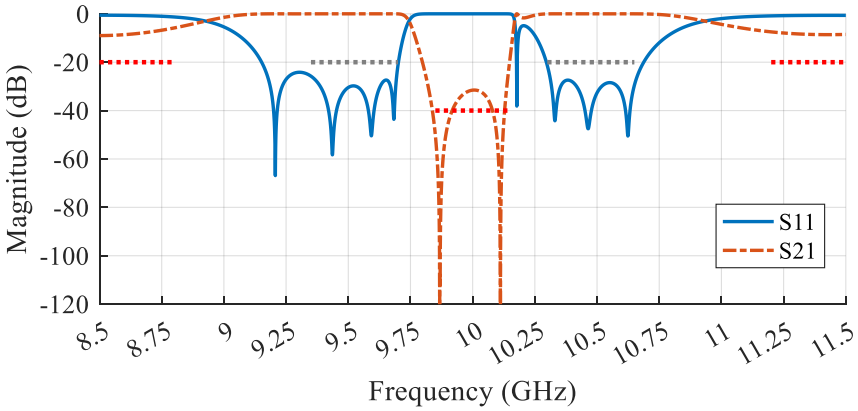
(a) Initial response



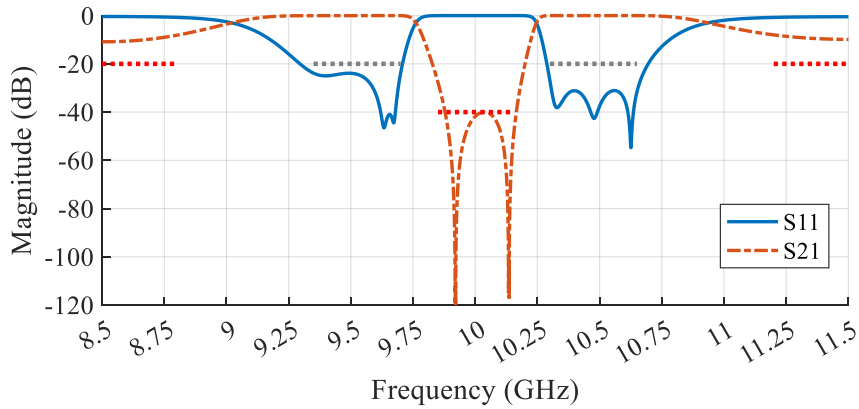
(b) Optimized response

Figure 3.8: Responses of the X-band filter using proposed methodology. (The grey dotted lines show the specification levels)

Five reference objective functions were compared. In all five runs, the ZPE objective function, the CM-difference objective function, and the cognition-driven multi-feature objective function fail to achieve successful results for this example. A typical response of the best design obtained using the ZPE objective function is shown in Figure 3.9(a), verifying that, as stated earlier, the ZPE objective function does not align with the design specifications after the general response shape is established. Similarly, the cognition-driven multi-feature objective function focuses on the correct positions for the zeros and poles and the ripple height of the passband, as shown in Figure 3.9(b), but fails to meet stopband and bandwidth requirements, even though the overall response shape is correct.



(a) An optimized response using ZPE objective function



(b) An optimized response using cognition-driven multi-feature objective function

Figure 3.9: Comparison of responses using different objective functions. (Grey dotted lines indicate met specs; red dotted lines indicate violations.)

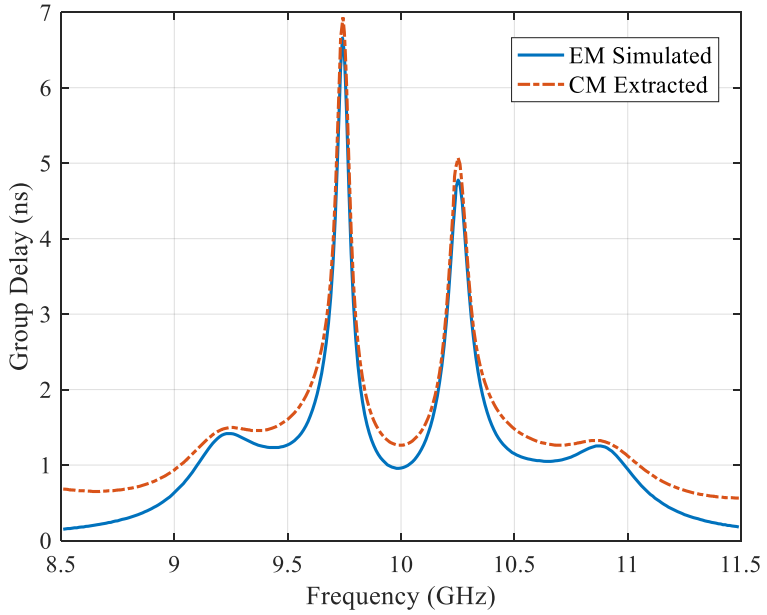


Figure 3.10: A group delay deviation using the CM-difference objective function.

For the CM-difference objective function, the result indicates a main challenge in accurately extracting the coupling matrix via group delay for many cases, leading to optimization errors, as shown in Figure 3.10. The group delay of extracted CM by a global optimizer (i.e., DE) still shows a large deviation compared to the simulated result. It's important to note that these functions, though ineffective for unsupervised design cases in this chapter, have shown success in semi-supervised design with some human intervention, as mentioned in the literature review.

The $\max(|S_{11}|)$, sole F_2, and hybrid $F_1 + F_2$ were all successful in the five runs, though their performance varied. The $\max(|S_{11}|)$ yielded successful results in three out of five runs. In contrast, both F_2 and the hybrid $F_1 + F_2$ achieved success in all five runs. The comparison results are summarized in Table 3.3, where only the successful runs for $\max(|S_{11}|)$ are included in the statistical calculations. It can be observed that the hybrid $F_1 + F_2$ significantly improves efficiency, reducing the number of EM simulations by 30% to 50% compared to the reference methods, based on average values. Additionally, the hybrid method has a much smaller standard deviation, indicating more stable and consistent performance.

Table 3.3: Statistical results for different objective functions using hybrid optimization algorithm

	$\max(\|S_{11}\|)$	Sole F_2	Hybrid $F_1 + F_2$
Success Rate	3/5	5/5	5/5
Min. Number of EM Sims.	1179	526	646
Max. Number of EM Sims.	1430	1298	740
Ave. Number of EM Sims.	1286	955	678
Standard Deviation	129	390	36

Next, a comparison between the hybrid and nonhybrid optimization algorithms using the same objective function, $F_1 + F_2$, is conducted. As previously mentioned, with the same initial population and random seed, the convergence trends of the two algorithms differ when NM simplex is triggered. At that point, both algorithms have the same current best design and training data points, after which their search mechanisms are conducted separately. The results from a typical initial population (out of five) are presented, as all runs show the same conclusion. The corresponding convergence trends are shown in Figure 3.11. Using the overall constraint violation threshold of 0.1, the hybrid approach required an average of 654 EM simulations to converge for this

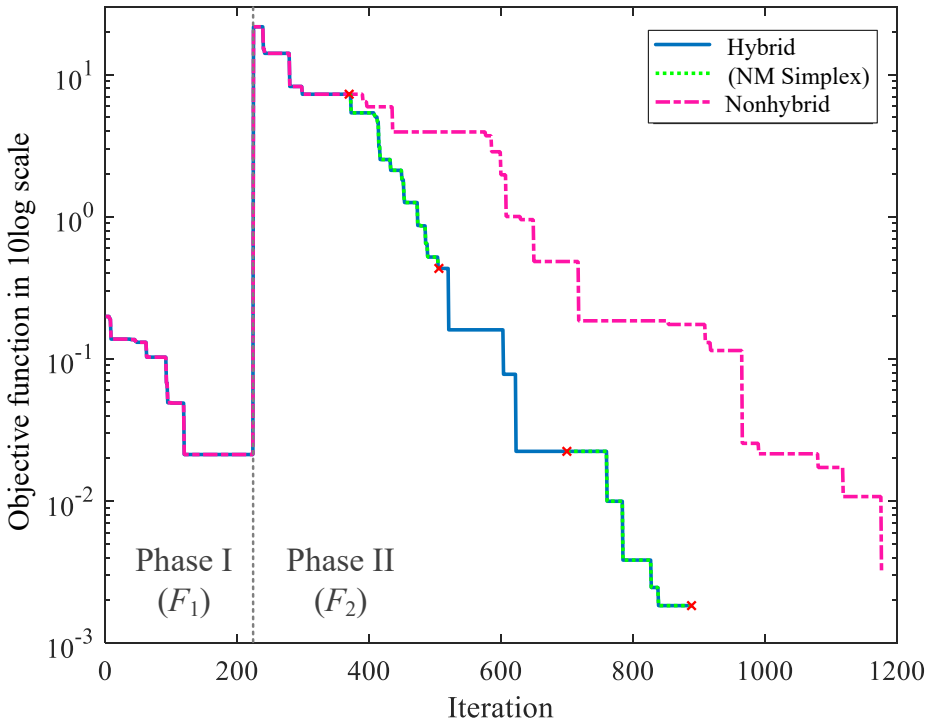


Figure 3.11: Typical convergence trends of the hybrid and nonhybrid optimization algorithms (example 1)

typical initial population, whereas the nonhybrid approach required an average of 1015 EM simulations across five runs. Thus, the hybrid algorithm reduces about 30% of the total EM simulations, demonstrating the improved efficiency of NM simplex and the combination of global and local search for optimizing the filter design landscape.

3.5.2 Example 2: C-Band Sixth-order Waveguide Filter

The second example is a C-band sixth-order waveguide filter with two transmission zeros, as shown in Figure 3.12. The operating frequency range is 4.9 to 5.1 GHz and resonators 2-5 form a cascaded quadruplet. In this filter, two transmission zeros are created at the upper and lower stopbands, resulting in higher stopband rejections. The resonators are coupled through inductive posts or coupling irises, as capacitive coupling irises are unsuitable due to the narrow passband frequency range. Consequently, the filter is designed using an H-plane cut structure. Resonator 4 employs a TE102 mode to achieve negative coupling between resonators 2 and 5.

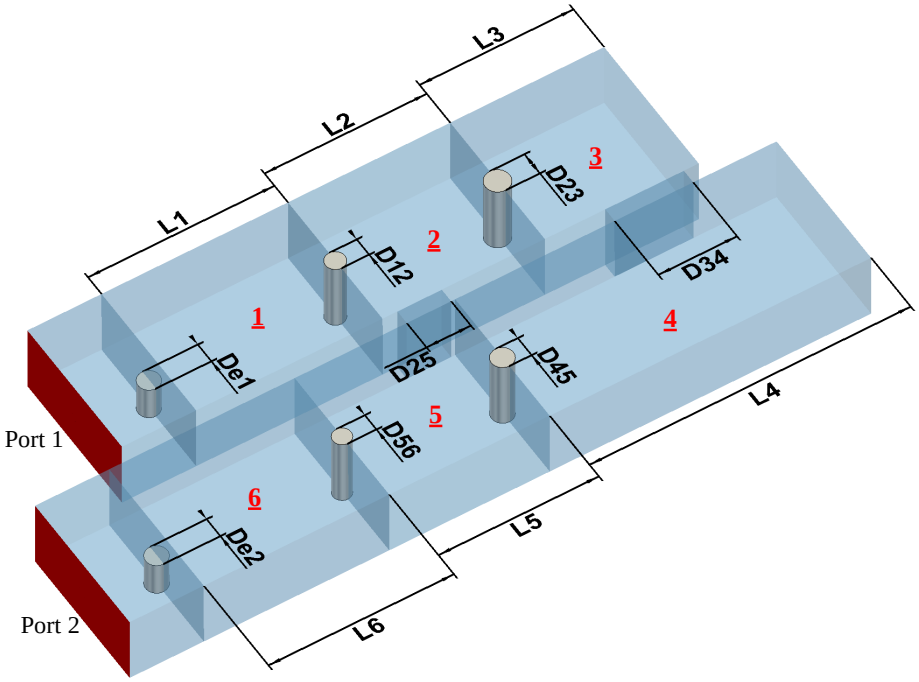


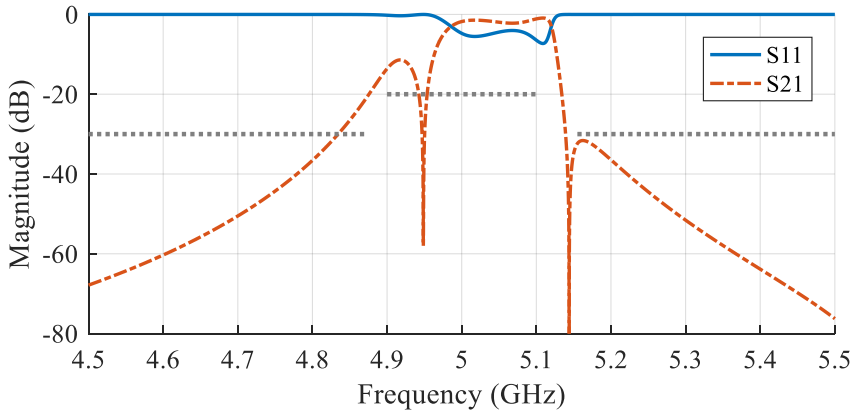
Figure 3.12: The structure of the C-band filter

This filter has 14 design variables, in which $[L1, L2, L3, L4, L5, L6]$ target resonant frequency and $[D12, D23, D34, D45, D56, D25, De1, De2]$ control the coupling between resonators. The filter is modeled in CST Microwave Studio with approximately 12000 mesh elements, and each EM simulation takes about 1 to 1.5 minutes. The design specifications are listed in Table 3.4, with notations marked in Figure 3.13(b).

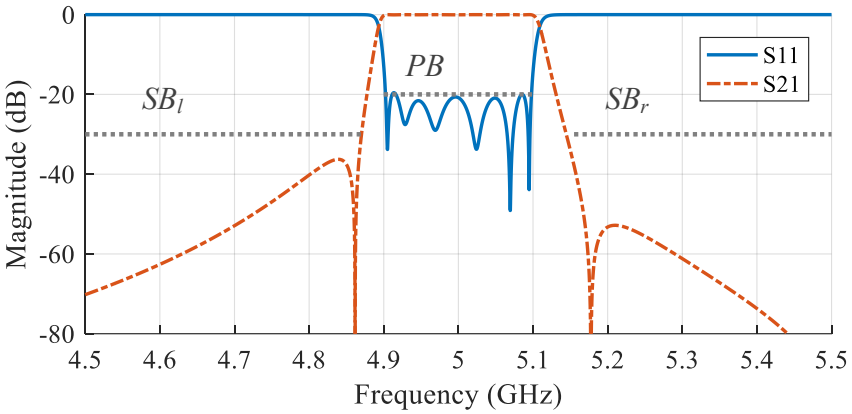
Table 3.4: Design specifications for example 2

Notation	**Item**	**Frequency Range** (GHz)	**Specification** (dB)
PB	Passband Reflection Coefficient (S_{11})	4.9 - 5.1	-20
SB_l	Stopband Left Edge Transmission Coefficient (S_{21})	≤ 4.87	-30
SB_r	Stopband Right Edge Transmission Coefficient (S_{21})	≥ 5.15	-30

In all five runs, the optimal design obtained after the two-phase optimization process successfully meets all specifications. The initial design and a typical optimal design are shown in Table 3.5, with the response displayed in Figure 3.13. Across these five runs, an average of 776 EM simulations were executed, costing 16 hours. As concluded in Example 1, this time consumption is reasonable considering the unsupervised design process.



(a) Initial response



(b) Optimized response

Figure 3.13: Response of the C-band filter using proposed methodology. (The grey dotted lines show the specification levels)

As for the comparison with five reference objective functions, the ZPE objective function, the CM difference objective function, and the cognition-driven multi-feature objective functions all failed to achieve successful results for this example, for reasons similar to those discussed in Example 1. The $\max(|S_{11}|)$ objective function also failed to find a design that satisfies all specifications. The optimization was found to be trapped in a local optimum with a maximum passband $|S_{11}|$ of -17.9 dB. A potential explanation is that, as discussed in the methodology, with the increasing number of orders of the filter, the edges of the passband response become steeper making the $\max(|S_{11}|)$-based objective function lead to a more complex design landscape [90], which causes the search to fail.

Table 3.5: The initial and a typical optimized design (all sizes in mm) (example 2)

Variable Name	$D12$	$D23$	$D34$	$D45$	$D56$
Initial Value	2.616	3.233	18.830	3.233	2.616
Optimized Value	2.509	3.246	21.230	2.919	2.325
Variable Name	$D25$	$L1$	$L2$	$L3$	$L4$
Initial Value	12.484	50.004	43.615	41.343	86.963
Optimized Value	12.101	50.006	43.310	41.320	86.568
Variable Name	$L5$	$L6$	$De1$	$De2$	
Initial Value	42.976	49.971	2.918	2.919	
Optimized Value	43.023	50	2.481	2.492	

Table 3.6: Statistical results for different objective functions using hybrid optimization algorithm

	Sole F_2	Hybrid $F_1 + F_2$
Success Rate	3/5	5/5
Min. Number of EM Sims.	1460	511
Max. Number of EM Sims.	2350	925
Ave. Number of EM Sims.	1916	776
Standard Deviation	445	168

Three out of five runs using the sole F_2 objective function were successful. The comparison results are shown in Table 3.6, where only the successful runs for F_2 are included in the statistical analysis. The results show that the hybrid $F_1 + F_2$ significantly improves the efficiency, reducing the required simulation by 60% compared to using F_2) alone. Additionally, the hybrid $F_1 + F_2$ has a much smaller standard deviation than using only F_2, demonstrating the benefits of F_1 and the combination of $F_1 + F_2$.

Comparisons between the hybrid and nonhybrid optimization algorithms show the same observation as in Example 1. All the five initial populations conducted similar results. For a typical initial population among the five, the hybrid algorithm reduces the total EM simulations by 30% compared to the nonhybrid algorithm, based on an average of five runs. The corresponding convergence trends are shown in Figure 3.14, where the hybrid approach takes 712 EM simulations, while the nonhybrid approach takes 1211 EM simulations. In this case, NM simplex optimization converged directly to the final optimal design when it was triggered. Once again, the effectiveness of the NM simplex optimization and the combined global and local search mechanism for filter design optimization is thus demonstrated.

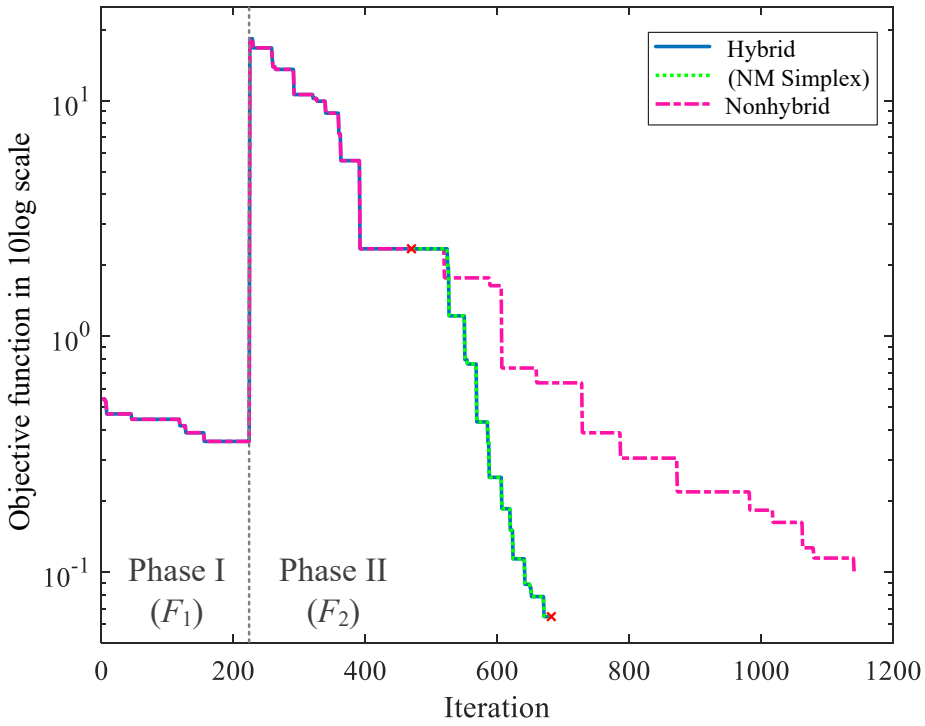


Figure 3.14: Typical convergence trends of the hybrid and nonhybrid optimization algorithms (example 2)

3.6 Summary

This chapter focuses on microwave filter design automation. It begins by discussing the general design process, identifying design optimization as the main challenge and bottleneck in achieving a higher degree of automation. A thorough review of recent literature reveals a lack of unsupervised design methodologies for general types of filter design optimization (beyond direct-coupled filters) without human intervention. The proposed end-to-end unsupervised design methodology combines a systematic sampling approach with a two-phase optimization process, utilizing two novel objective functions and a hybrid optimization algorithm. Its effectiveness is demonstrated through the design of two examples: an X-band dual-band filter and a C-band sixth-order waveguide filter. Comparative analyses show the superiority of the proposed objective functions and algorithms. The process can be completed in approximately half a day on a standard desktop computer without decision and intervention from designers. These real-world examples, which were previously considered challenging for unsupervised design with existing methods, demonstrate that the proposed methodology enables unsupervised filter design automation with reasonable efficiency.

Chapter 4

MMIC Power Amplifier Design Automation

4.1 Background

The growing demand for high data-rate communication systems has driven the rapid development of RF and millimeter-wave (mm-wave) technology, targeting applications such as 5G or satellite communications and remote sensing. Power amplifiers (PAs) are one of the critical components of RF and mm-wave front ends, responsible for amplifying the power of small RF signals radiated through antennas. As the sole nonlinear large-signal device in both downlink and uplink, PAs have significant impacts on system performance, influencing factors such as data rate, latency, and more. Different systems have different performance requirements; for instance, a radar application requires high power and high efficiency, while a base station application favors good linearity and wideband flat gain response. Consequently, there is a lack of straightforward, universally applicable procedures for PA design, though many PA design methods and configurations have been proposed in recent decades, e.g., Class F [125], Doherty [126], and Outphasing [127].

To delve into this issue, the conventional PA design methodology is reviewed. While the performance specifications (e.g., gain, efficiency, or output power) are defined according to the application scenario, the transistor technology (e.g., LDMOS, GaAs, or GaN) is first determined to ensure a sufficient margin of performance. Subsequently, the configuration of PA is settled, and based on the corresponding design theory, the topology and the preliminary ideal design are drawn out with lumped components.

Due to the non-linear operations of PAs, the input and output impedance must be carefully selected through load- and source-pull experiments or simulations. Note that the target output impedance may vary depending on different power levels (e.g., for Doherty PAs at full power and backoff) and frequencies (e.g., wideband PAs), requiring one matching network that serves multiple purposes, which significantly increases the design complexity. Once the ideal design is optimized and shows acceptable responses, the lumped elements are then replaced by distributed or model-based components (e.g., transmission lines and MIM capacitors). Manual trail-and-error tuning is then performed on schematics, aiming to achieve an optimal design that meets all requirements. Due to computational burden, EM simulations are often performed only at the end of the design procedure before or around layout. If the performance deteriorates undesirably in EM simulation, which is common in mm-wave PAs, designers must conduct manual fine-tuning or apply local search (often via built-in optimizers in EDA software) for further optimization.

This situation becomes even more complicated when considering multistage PAs, which include both driver stages and a final stage, as often seen in monolithic microwave integrated circuits (MMICs). MMIC PAs integrate both driver and final stages into a single chip, offering a compact and cost-effective solution, especially for mm-wave band operations, This potential has attracted considerable attention in recent decades due to the development of 5G and 6G communication. To design MMIC PAs, specifications are first decomposed into several requirements for each stage. Each stage is then designed separately using the aforementioned process and combined for layout, followed by manual fine-tuning and optimization. While this approach is manageable and heavily relies on the designer's experience, achieving an optimal design remains challenging due to the intricate correlations between parameters and responses, even for seasoned engineers.

Thanks to the rapid development of computing power and the reduction in its cost, fully optimization-oriented automated PA design methods have emerged as a crucial innovation to enhance efficiency and performance in PA development [128]. Without considering EM simulations, several successful optimization-oriented design algorithms working at the schematic level (including only circuit simulations with lumped elements) have been proposed, demonstrating excellent performance for various sub-6GHz PAs [129, 130, 131, 132]. These methods, employing global optimization algorithms (e.g., simulated annealing, brainstorm optimization, and particle swarm optimization), optimize parameters of board-level one-stage PAs (i.e., a discrete transistor with its matching networks), often resulting in commendable designs. However, for high-frequency and

intricate applications, particularly considering microwave and MMIC PAs, the performance of schematic-level designs often degrades sharply after rendering EM simulation. Thus, performing PA design optimization at the layout level is often necessary, and remains a significant challenge.

This challenge becomes more evident when dealing with many parameters (e.g., around 30 to 50) and the need for consistent performance (i.e., good flatness of efficiency, output power, and gain over frequency) across the entire band for multiple stages. Considering the optimization-oriented PA design at the layout level, the computational cost of EM simulations is a critical factor, making the direct use of global optimization algorithms prohibitive.

A possible approach to mitigate this issue is by constructing offline ML models as proxies in lieu of costly simulations. In this approach, parameters of components or reusable structures are sampled in advance and simulated extensively. As a result, ML models are trained to capture the correlation between parameters and performance (e.g., power, gain, or efficiency). By applying these computationally cheaper surrogates, modern global or multi-objective optimization algorithms can be employed without sacrificing efficiency. Successful results are shown in [131] and [133]. However, this kind of method is often suitable for PAs with plain topology (i.e., one stage with only bias and matching networks) or simply cascaded matching networks, and less stringent specifications, e.g. a typical one-stage class-AB PA or a transformer-based CMOS PA. When one of these factors increases in complexity, such as in multistage Doherty PAs, training accurate offline models for components or reusable structures becomes a new burden prior to PA design optimization.

In contrast to building offline machine learning models, another approach, capable of addressing more general and intricate cases, integrates machine learning with optimization algorithms, allowing ML models to be trained online for performance prediction. Bayesian optimization framework, detailed in Section 2.3, is employed in [134, 135, 136], which uses GPs incorporating various constrained and multiobjective optimization algorithms. Several sub-6GHz broadband and Doherty PAs with medium-scale design variables were successfully designed. However, while these methods and studies offer promising directions, they still have limitations in terms of efficiency, effectiveness, or generality, which will be detailed in the next section.

The discussion above highlights the significant demand and necessity for investigating holistic machine learning-assisted design methodologies that facilitate the synthesis of PAs with stringent performance requirements, considerable complexity, and high integration density, while achieving a higher degree of automation. Specifically, there is a critical need for PA design optimization at the layout level that can eliminate the need for manual tuning. In the following content, several selected papers are reviewed in detail and their pros and cons are analyzed respectively. Then the main problem is clarified, and the research goals are emphasized. Finally, the proposed methodology is explained in detail, followed by experiments and discussions.

4.2 Literature Review

Table 4.1 summarizes publications in recent decades related to PA design automation, which can be categorized from various perspectives. In this review, we classify these works into four categories based on the algorithms and ML techniques employed. The first category (I) includes papers [137] and [138], which utilize a rule-based circuit matching technique, i.e., simplified real frequency technique (SRFT), to produce and optimize theoretical or ideal PA designs. These methods are effective for assisting experienced engineers in designing from scratch, but the resulting designs are often far from the final outcomes, which therefore is not the main focus. The second category (II) includes papers [130, 133, 139, 140, 141, 142, 143], which typically employ ad hoc global optimization algorithms for schematic design, mainly applied to discrete PAs at board level, and generally excluding ML techniques. The third category (III) features papers [131, 144, 145], which integrate offline ML with optimization algorithms for PA design. The forth category (IV) includes paper [134, 135, 136, 146, 147, 148], which employ online ML with optimization algorithm. Several key papers selected from the (II), (III), and (IV) are discussed in the following context.

Two representative studies in category (II) are [129] and [130]. In these studies, the authors utilized a simulated annealing particle swarm optimization (SA-PSO) method for designing high-efficiency and broadband discrete PAs. As a heuristic optimization algorithm, SA-PSO can find excellent solutions by directly optimizing drain efficiency and output power, without requiring redundant source/load-pull simulations and manual tuning, offering the main strength of these methods. However, these methods are limited to schematic-level design optimization, of which circuit simulation results can be

Ref.	Date	Algorithm	ML Tech.	Design Level	Objective Func.	Sim. Budget	PA Type
[139]	Dec. 2012	PSO	None	Schematic	Impedance	2000-5000	Discrete
[137]	May 2013	SRFT	None	Ideal	Reflectance	-	Theoretical
[146]	Feb. 2014	SAEA	GP	Schematic, EM	PAE, Power, Gain	∼1000	CMOS IC
[140]	Sep. 2014	GA	None	Schematic	Impedance	-	Discrete
[138]	Oct. 2014	SRFT	None	Ideal	Reflectance	-	Discrete
[134]	Dec. 2015	BO	GP	Schematic, EM	Harmonic power	1000-40000	Discrete
[141]	Jan. 2017	GA	None	Schematic	Impedance	-	Discrete DPA
[135]	Mar. 2017	BO	GP	Schematic, EM	Impedance	2000-20000	Discrete DPA
[147]	Aug. 2018	BO	GP	Schematic	PAE, Power, Gain	5000	Discrete
[149]	Nov. 2019	BO	-	Ideal to Schmetic	Impedance	-	Discrete
[142]	Dec. 2019	∼PSO	None	Schematic	Performance	3000	CMOS
[131]	July 2020	TSEMO	Offline DNN	Topology, Schematic	Gain, PAE	5000	Discrete
[144]	May 2021	-	Offline DNN	Ideal	Eff., Gain, Power	-	Discrete
[129]	May 2021	SA-PSO	None	Schematic	Power, Eff.	8000	Discrete
[148]	May 2022	BO	GP	Schematic	PAE, Power	-	Discrete
[136]	Aug. 2022	BO	GP	Schematic	Eff., Power	1000	Discrete DPA
[150]	Aug. 2022	MOBO	GP	Schematic	Harmonic power	10000	Discrete
[130]	Sep. 2022	SA-PSO	None	Schematic	Eff., Power	-	Discrete LMBA
[143]	Nov. 2022	NSGA-II	GP	Schematic, EM	PAE, Power	-	Discrete
[145]	July 2023	-	Offline DNN	Topology, Schematic	Power, Reflectance	-	Discrete LMBA
[133]	July 2023	NSGA-II	Offline	Schematic, EM	S-para, Power,PAE	∼3000	CMOS IC

Table 4.1: Published research about PA design optimization in recent decades.

- : The item is not given or not mentioned in this article.
None : The item is not necessary in this article.

obtained quickly, often within seconds. To quantify this limitation, the maximum number of simulations, or the simulation budget (estimated by multiplying the number of iterations and the population size) was set to 8000 in the experimental cases discussed. Their time cost becomes prohibitively expensive when considering EM simulations.

In category (III), an innovative DNN-based PA design technique was presented in [131], which can automatically determine the topology of matching networks and estimate the value of components as well. This method uses an offline, pre-trained DNN—specifically, long short-term memory (LSTM) neural networks—as a classifier to predict the number and the corresponding topology of lump components, and a regression model with Thompson sampling optimization to optimize their values. While this method indeed reduces the need for manual tuning, it must be deployed on a case-by-case basis, as the DNN models need to be retrained from scratch for different PA configurations. For instance, each time the classifier determines a topology, the regression models must be re-trained based on that specific topology. Additionally, training high-quality offline DNN models requires numerous simulations and careful parameter adjustments, making the process complex and time-consuming for general cases.

In category (IV), the key work proposed in [134] introduced an optimization-oriented PA design method based on BO framework. In this method, PA performance is optimized indirectly by optimizing the drain waveforms. This method is verified using both schematic and EM simulations, resulting in excellent solutions within a simulation budget of 1000 (with EM simulation) to 4000 (with only schematic simulation on lumped elements). The authors later extended this work to a multi-objective case aiming to broadband high-efficiency PAs [135], where the mean square error of the target and tested impedance was optimized to achieve the desired performance. This method is viable for PA design optimization within a few thousand simulations and has been shown to outperform built-in EDA optimizers; however, it is not intended for engineers without experience, as the objective function is not straightforward and needs to be defined on a case-by-case basis. Furthermore, due to the lack of search operators in BO, an initial high-quality design is often required, as its quality significantly impacts the final outcome.

Another key research in category (IV) is in [146], where a surrogate model-assisted IC synthesis method called GASPAD was proposed and validated using two 60GHz CMOS PAs. This method employs the SAEA framework, combined with GP models to predict performance and reduce the need for EM simulations during optimization. To bal-

ance the efficiency (convergence speed) and exploration (optimization ability), a model management method called the surrogate model-aware search mechanism (SMAS) is utilized. This method is innovative in terms of its algorithm, with the search engine being carefully analyzed and designed. However, it primarily targets CMOS ICs with a narrow bandwidth and simple typologies (cascaded transformers with CMOS transistors). When they are applied to complicated PAs (e.g., MMIC Doherty PAs) with stringent specifications, achieving satisfactory results can be very challenging.

In conclusion, while the paper discussed above attempts to tackle the problem of PA design automation from various perspectives, particularly for those in category (IV), their effectiveness remains limited. First, most methods require thousands to tens of thousands of simulations to achieve the desired design, which makes them impractical for complex real-world applications. Second, many methods lack consideration of both effectiveness and generality during their development. The use of specific objective functions, such as harmonic power, or impedance error, does not necessarily guarantee that the required specifications about overall power output and efficiency are met, greatly limiting their applicability. In the next section, the main problem to address is further clarified and described before introducing the proposed methodology.

4.3 Problem Description

Building on the previous discussion, this research aims to propose a new PA automated design methodology that operates at the layout level, i.e., including both large-signal circuit simulation and EM simulation for holistic characterization before tape-out. Given the circuit topology, design variables, and specifications, the proposed methodology can synthesize satisfactory PAs through a fully optimization-oriented approach. Note that this work does not focus on synthesizing or modifying the circuit topology (such as adjusting, adding, or removing components from a given circuit) during design optimization, although this remains an appealing challenge to address.

Additionally, the proposed methodology is specifically designed for MMIC PAs. Compared to board-level discrete PAs, MMIC PA design is more intricate in the following aspects:

- As previously mentioned, MMIC PAs are often multistage, involving more complex design processes, numerous design parameters, and stringent specifications. Therefore, a methodology proposed for MMIC PAs can seamlessly be applied to board-level discrete PAs, but not vice versa.
- MMICs often have a more complicated layout with many folds and bends in transmission lines due to the space constraints of chips. In contrast, board-level PAs are often laid out by sequentially connecting several matching components without significant space considerations.
- Board-level PAs can often be adjusted post-fabrication to address possible deviations against expectation, especially for sub-6 GHz applications. In contrast, MMICs, frequently operating at microwave and mm-wave frequencies, are challenging to tune manually after tape-out. Hence, ensuring a high first-pass success rate through comprehensive design and optimization becomes crucial for MMICs.

Given these considerations, it is reasonable to expect that a methodology proposed for MMIC PAs can also work well with other PAs and MMICs. To achieve this, the following goals are considered essential:

- Compatible with the current MMICs design environment. The design of MMICs relies on process design kits (PDKs) provided by foundries. PDKs typically include structures and simulation models of components based on a specific technology and are integrated with EDA software (e.g., Advanced Design System (ADS)). To ensure accurate characterization of MMICs through various simulators (e.g., harmonic balance, S-parameters, and momentum) and to allow seamless transition from upstream topology or initial schematic design, the methodology must be compatible with the existing design environment.
- General enough to be applicable to different PA configurations. The methodology should be independent of PA configurations and should be straightforward to use by directly optimizing explicit performance rather than specific design metrics, such as harmonic power consumption, overlap of drain current-voltage waveforms, impedance-matching errors, and so on.
- Capable of handling multistage PAs with stringent specifications. A typical MMIC PA often includes two or three stages, resulting in 30 to 50 design variables, and requires optimization across approximately 10 specifications covering small-signal (e.g., return loss, gain, and gain ripple) and large-signal (e.g., power-added efficiency and output power) performance.

4.4 Proposed Methodology

To achieve the aforementioned goals, the proposed methodology consists of two fundamental components: the optimization-oriented integrated environment and the BNN-based optimization algorithm. Both are essential for achieving a layout-level synthesis of MMIC PAs. As the most popular MMIC design software, Advanced Design System lacks a built-in interface for communication with external programming languages (as of this thesis is writing). Therefore, the optimization-oriented integrated environment is implemented as infrastructure to bridge the programming environment and simulation environment, streamlining both dataflow and workflow. The BNN-based optimization algorithm, which is the core of the proposed methodology, is designed to adapt to the landscape of PA design problems, achieving a good balance between exploration and efficiency. Each block is explained in detail below.

4.4.1 Optimization-oriented Integrated Environment

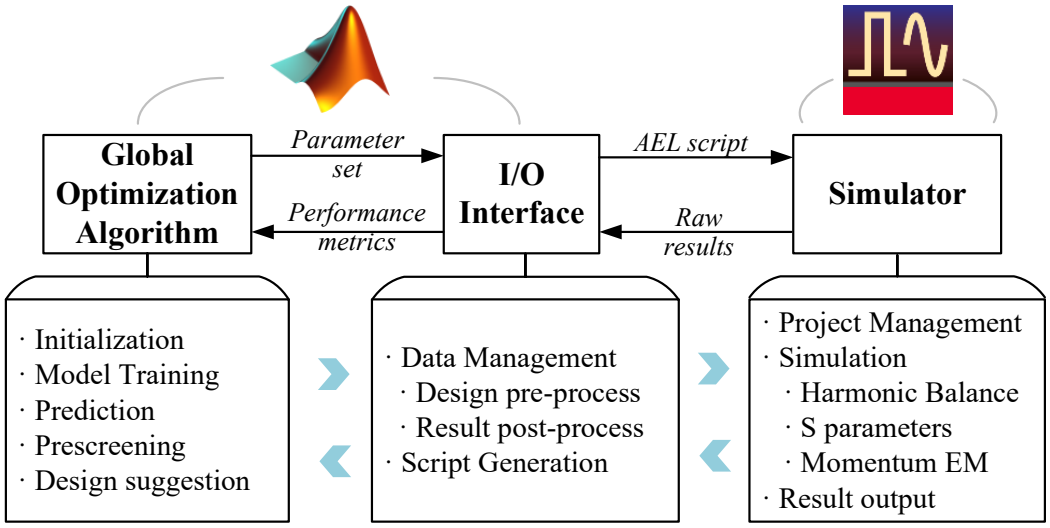


Figure 4.1: Workflow of the proposed optimization-oriented integrated environment.

As the name indicates, the optimization-oriented integrated environment bridges the optimization algorithm (implemented by MATLAB) with simulation software. Figure 4.1 illustrates this workflow, and depicts the interaction between three key elements: the algorithm, the input/output (I/O) interface, and the simulator, along with their corresponding functions. The optimization algorithm manages the entire optimization process, including design initiation, model training, performance prediction, design prescreening, and so on. When the algorithm suggests a new parameter set, the I/O interface converts these values into a valid format recognizable by EDA soft-

ware, ensuring they are correctly formatted with appropriate precision and units (i.e., design pre-process). An AEL (application extension language) script is then generated to control the simulation software. Once the script execution is complete, the raw performance results are retrieved and abstracted into several performance metrics, which are then fed back and provided to the optimization algorithm for further processing (i.e. result post-process).

AEL is a built-in programming language in ADS, designed to record and automate manual operations within the main window. Since AEL is not officially designed to be called externally through command line, assistance from operation system is often necessary. An example code is provided in Appendix B. Generally, ADS hosts and manages the whole design project with PDKs, while the AEL script adjusts parameters within the design project, runs, monitors a series of simulations, and exports raw performance data.

In addition, for a holistic layout-level characterization of MMIC PAs, simulation involves several steps. First, geometric structures of passive components (i.e., microstrip lines, capacitors, etc.) are constructed and simulated using momentum simulator. Next, S-parameters involving active components (i.e., transistors and power source) are simulated to obtain small-signal performance, such as input and output return loss and gain. Finally, harmonic balance (HB) simulation is performed to obtain nonlinear large-signal performance, such as amplitude modulation to amplitude modulation (AMAM), and drain efficiency. However, the HB method, which assumes steady-state solutions can be approximated with a finite Fourier series for a given sinusoidal excitation, is iterative and may suffer from convergence issues. To diminish the influence of failed convergence of HB simulation, a retry mechanism is implemented within the AEL script. If the simulation still fails after several retries, an error flag is passed to the I/O interface, and a high penalty value is assigned.

To the best of the author's knowledge, this optimization-oriented environment, which is compatible with most PDKs provided by foundries and integrates both EM and large-signal simulations for MMICs, is proposed and implemented for the very first time. This distinguished the proposed method from others mentioned in the literature review, enabling a higher degree of PA design automation.

4.4.2 BNN-based Optimization Algorithm

Although several prospective studies have employed Bayesian optimization as the framework for PA design optimization [134, 135], it is not well-suited to the problem described in section 4.3. Some of its limitations have been discussed in section 2.3. First, BO lacks a guaranteed optimization engine (the engine that efficiently searches the posterior distribution within an acquisition function), which significantly reduces the algorithm's efficiency. Second, the machine learning technique used in BO, i.e., GP, suffers from high computational complexity, especially when modeling multiple performance metrics (around 10 in the case of the experiment section) separately. Consequently, experimental discussion in [134] suggest that their method often relies on a high-quality initial design, with the search range typically restricted to $\pm 10\%$ around this design which greatly limits their potential to find optimal solution.

To address these issues, the proposed algorithm utilizes SAEA as the main framework because of its well-known global optimization capability, and leverage DE operation as the search engine. BNNs are employed in place of GPs to model PA performance metrics during optimization process. As mentioned in Section 2.2.4, computational costs of BNNs are significantly reduced compared to GPs. In addition, the model management method proposed in [146], i.e., SMAS, is adapted and modified to strike a balance between algorithm efficiency and exploration of PA design solutions. The pseudo-code of the proposed BNN-based optimization algorithm is shown in Algorithm 8.

The algorithm does not rely on an initial layout design and only uses a schematic design with transmission lines. Therefore, it begins by sampling a set of designs using a Latin hypercube sampling method within the given design space, ensuring that the initial dataset $\mathcal{D}$ is well-distributed. This approach eliminates the need for a high-quality initial design, requiring only a defined search range. Note that the initial design and search range are provided and estimated based on the basic schematic simulation. And it is found negligible impact to the performance of the algorithm. Additionally, when any variable reaches the given search bound, the bound is expanded by 10% until the maximum allowed simulation value is reached. The simulation counter is then initialized to zero in Line 2, and all designs are sorted in ascending order according to a penalty function. The penalty function penalizes every designs that violate any specifications. When the penalty function value is zero, all specifications should be satisfied. Subsequently, the top N solutions are selected from the sorted designs to form the initial population.

Algorithm 8 BNN-based Optimization Algorithm

Input: Design dataset $\mathcal{D}$, penalty function $F(\cdot)$, population size N, simulation program $\texttt{Sim}(\cdot)$, maximum number of simulations M

1: Initialize database $\mathcal{D}$ by Latin hypercube sampling
2: Set simulation counter $n \leftarrow 0$
3: Sort all designs in $\mathcal{D}$ in ascending order by $F(\cdot)$
4: Select top N solutions from $\mathcal{D}$ to form the population $\mathcal{P}$
5: **repeat**
6: **if** n is odd **then** ▷ Local DE search
7: Perform local mutation on $\mathcal{P}$ with a scalling factor F_l
8: Perform crossover operator to have $\mathcal{P}_o$
9: **else** ▷ Global DE search
10: Perform global mutation on $\mathcal{P}$ with a scalling factor F_g on $\mathcal{P}$
11: Perform crossover operator to have $\mathcal{P}_o$
12: **end if**
13: Train BNN models with parameter inheritance for each solution in $\mathcal{P}_o$
14: Predict and prescreen solutions in $\mathcal{P}_o$ by BNN models
15: Select the best $\boldsymbol{x}_o \in \mathcal{P}_o$ with the minimium prescreening value
16: Simulate $\boldsymbol{x}_o$ and add $\boldsymbol{x}_o$ with simulation result $\texttt{Sim}(\boldsymbol{x}_o)$to $\mathcal{D}$
17: Set simulation counter $n \leftarrow n + 1$
18: Sort all designs in $\mathcal{D}$ in ascending order by $F(\cdot)$
19: **if** n is odd **then** ▷ Population reconstruction
20: Select top $2N$ designs from $\mathcal{D}$ to form the population $\mathcal{P}$
21: Cluster $2N$ designs into N groups by k-means clustering method
22: Remove the design with higher penalty value in each cluster
23: **else**
24: Select top N design from $\mathcal{D}$ to form the population $\mathcal{P}$
25: **end if**
26: **until** Stopping criteria are satisfied or $n > M$

Output: Best solution $\boldsymbol{x}_{\text{best}}$ and the corresponding penalty function value

In the main loop of the algorithm, a *hybrid* search engine is proposed, of which local and global DE searches are alternated. Specifically, the local DE search uses DE/best/1 mutation operator with a relatively small scaling factor F_l, focusing on improving the current best design by exploring nearby solutions. In contrast, the global DE search employs DE/current-to-best/1 mutation operator with a regular scaling factor F_g to maintain global search capability. The crossover operations (Lines 8 and 11) remain the same. Compared to standard SAEAs, the introduction of local DE search accelerates convergence. The underlying principle is that, given the dimensionality of PA design problems, a solution generated by DE/current-to-best/1 is likely to be further improved by exploring its neighborhood. When a new current-best design is found through local search, the entire population benefits in the subsequent global search iteration.

Unlike the standard local optimization algorithm (e.g., NM-simplex) used in many optimization problems (e.g., filter design optimization in Chapter 3), DE/best/1 with small scaling factor F_l is preferred due to its flexibility. This approach allows for a more flexible search process by adjusting the scaling factor, effectively exploiting better solutions within the promising region controlled by the scaling factor, whereas NM-simplex often fails to do so. Additionally, DE/best/1 is less likely to get trapped in local minima compared to NM-simplex, making it a robust choice for PA design problems.

After the offspring population $\mathcal{P}_o$ is generated, BNNs are trained for each solution within $\mathcal{P}_o$, which is different from the SMAS strategy used in [146]. Training data is selected by choosing the N nearest designs in $\mathcal{D}$, based on Euclidean distance, which helps to improve model accuracy. To further speed up the training process, the parameters of BNNs are passed down in each iteration. This means that, except for the first iteration, subsequent training builds upon previously learned parameters (the last posterior). Pilot experiments indicate that BNNs require 1/50 of the time needed by GPs for performance training and prediction in practical PA design problems. The trained BNNs are then used to predict and prescreen the solutions within $\mathcal{P}_o$, employing the LCB prescreening method. Only the best solution in $\mathcal{P}_o$ with minimum LCB value is simulated and added to $\mathcal{D}$ for the next iteration.

Furthermore, Lines 19 to 25 describe the process called population reconstruction. Due to the introduction of local DE search, the population diversity may decrease on some occasions. To counteract this effect, different reconstruction processes are employed. When the next iteration involves a local DE search ($n \leftarrow n + 1$ is odd), the top $2N$ designs are clustered and selected to reduce design similarity, enhancing diversity for the subsequent local search. Conversely, when the next iteration is a global DE search, the regular top N designs are selected, as they generally maintain sufficient population diversity. This careful balance between exploration and optimization speed has proven effective for PA design shown in the next section.

4.4.3 Numerical Experiment on Benchmark Problem

To assess performance of the proposed algorithm, numerical experiments on benchmark problems were conducted. Due to the computationally expensive simulations, it is difficult to verify the algorithmic performance of the proposed algorithm through practical PA design problems, as it requires many statistical runs. Therefore, a compu-

tationally inexpensive mathematical benchmark problem was used. Since the PA design landscape is often multimodal (i.e., containing many local optima) [129], a highly multimodal mathematical benchmark problem is chosen, which is the Griewank function with 20 variables, as formulated in Appendix A. The search range for each variable is [-600,600], with the known global minimum at $f(\mathbf{0}) = 0$.

A comparison was performed among three SAEAs: (a) the proposed algorithm, (b) SMAS with GPs, and (c) SMAS with BNN. SMAS with GPs is the core of GASPAD, while SMAS with BNNs replaces the GP modeling technique with BNNs. This comparison allows for observing the contribution of the introduced BNN modeling and the *hybrid* search engine. The computing budget, i.e., the maximum number of simulations, is set to 2000, and 20 independent runs were carried out for each method due to the stochastic nature of the algorithms.

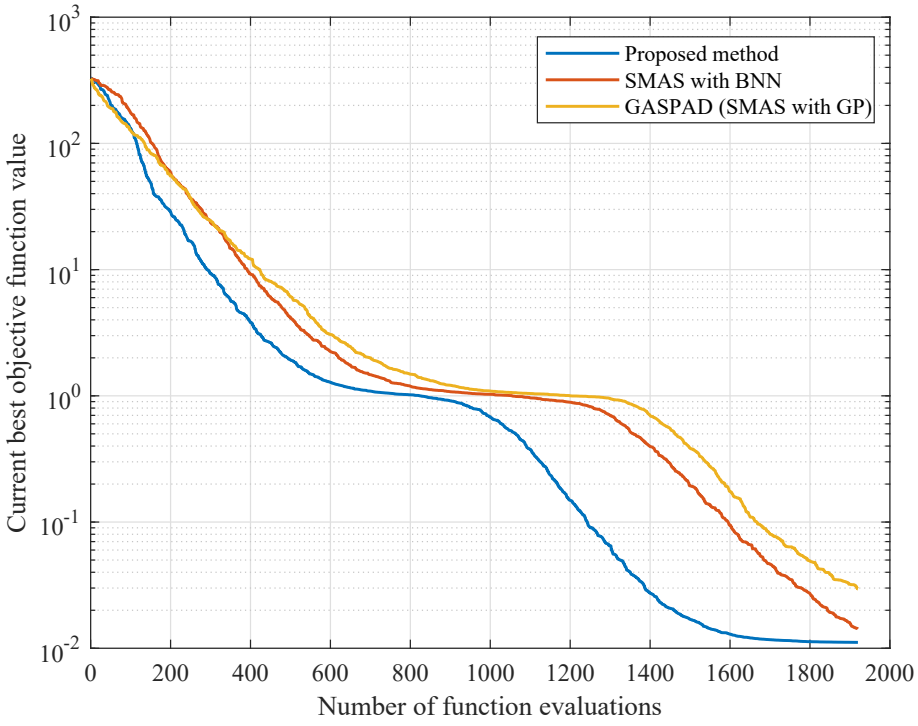


Figure 4.2: The convergence trends of the proposed method, SMAS with GP and SMAS with BNN

The statistical results are summarized in Table 4.2. Figure 4.2 shows the convergence trends. In terms of the statistical results, the proposed algorithm achieves the best average and median values. While SMAS with BNNs underperformed compared to the proposed algorithm, it still outperformed SMAS with GPs, demonstrating the benefits

of introducing both BNNs and the hybrid optimization engine. As for the convergence curve, a significant improvement with the proposed algorithm is observed, primarily due to the local DE search and enhanced modeling accuracy achieved through individual-wise BNNs.

Table 4.2: Statistical results on the benchmark problem

Method	Min (Best)	Max (Worst)	Mean	Median
SMAS with GPs	0.006534	0.105479	0.029456	0.024908
SMAS with BNNs	0.002238	0.031217	0.014240	0.015774
Proposed algorithm	0.000014	0.034589	0.011134	0.008688

4.5 Experiment

In this section, two real-world PAs are used to demonstrate the proposed methodology. The first example is a 27-31 GHz class-AB PA designed for satellite application, and the second example is a 24-31 GHz wideband Doherty PA for both 5G mm-wave base station and satellite application. Both cases utilize GaN-on-Si 100 nm technology. The schematic topologies for both examples are designed using PA synthesis theory [151, 152] with a wide search range for design variables provided by the designer. As mentioned earlier, layout-level design optimization is carried out, and the holistic characterization is conducted. To compare the performance of the proposed algorithm, GASPAD [146], i.e., SMAS with GPs as mentioned in above section, is implemented as the reference method. Both algorithm incorporates with the implemented optimization-oriented environment. As noted in the literature review, GASPAD is the only available method targeting general PA specifications and IC design with EM simulations.

Additionally, to prevent component overlaps while adjusting parameters during optimization, two layout control strategies are used. First, components between each stage are grouped into several functional sub-circuits, such as bias networks or matching networks. Each sub-circuit is then simulated by momentum as an independent unit. Second, when the geometric constraint of the layout (e.g., the chip die must fit within 2mm $\times$ 2mm) is imposed, transmission lines are pre-folded to maintain a specific region, and the design parameters are carefully set to preserve its shape. The detail of the pre-folding strategy is explained in the second example.

The penalty function used for optimization is formulated in 4.1, where y_i^{spec} is the i^{th} specification and y_i^{worst} is the worst value of the i^{th} specification during initial sampling (often the worst design during optimization as well). This step normalizes all specifications in a minimization direction, ensuring that when the penalty function value is zero, all specifications are satisfied.

$$F = \sum_i \frac{\max(y_i - y_i^{\text{spec}}, 0)}{y_i^{\text{worst}} - y_i^{\text{spec}}} \tag{4.1}$$

Due to the stochastic nature of the algorithm, random numbers may affect the performance outcomes. Hence, considering the computationally expensive simulations, four independent runs are conducted for each example with statistical results analyzed accordingly. For the reference method, the same four initial populations are used, and the four independent runs are performed. The experiments are run on an AMD Ryzen Threadripper PRO 3975WX 32-core workstation under the Linux operating system, with all the time measurements referring to wall-clock time.

4.5.1 Example 1: 27-31 GHz Balanced Class-AB MMIC PA

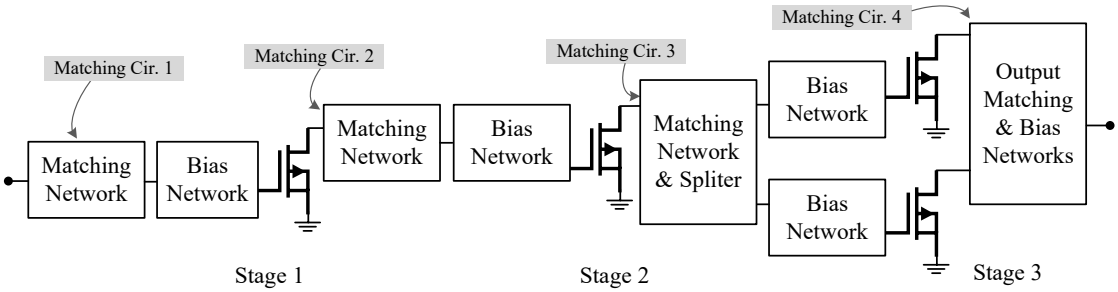


Figure 4.3: Top-level schematic of the balanced class-AB MMIC PA.

The first example is a balanced class-AB PA consisting of two driver stages and a final stage. The input signal is first amplified by the driver stages, and then split equally (so-called balanced) by a non-isolated divider, amplified in two branches, and combined at the output. This PA is designed to operate over 27-31 GHz, covering 5G FR2 bands n257 and n261. The main design challenge is to achieve consistent performance (i.e., gain, output power, and PA efficiency) across the entire bandwidth since multiple stages are involved.

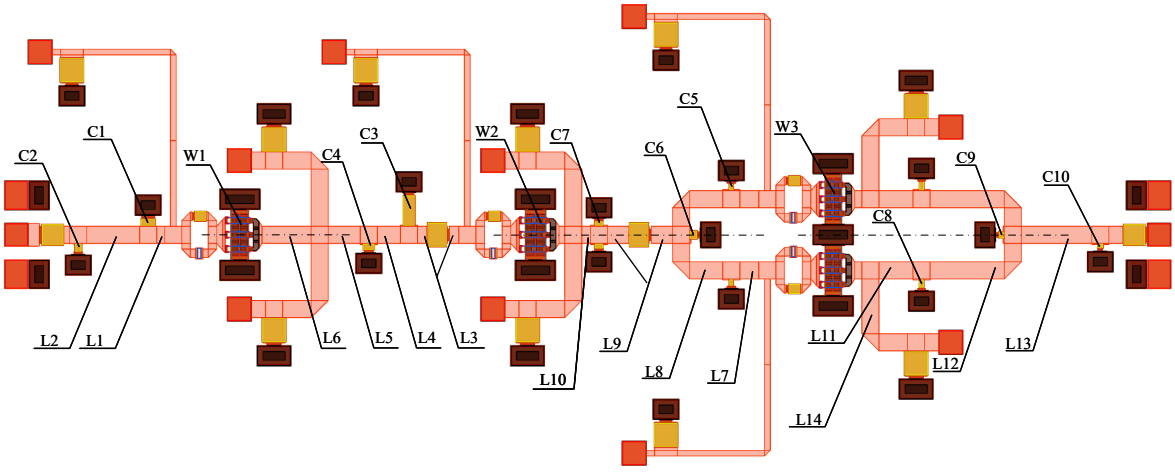


Figure 4.4: A layout photo of the balanced class-AB MMIC PA

Table 4.3: Design variables, search ranges, and a typical optimal design obtained by proposed method (Example 1)

Type	b	a	b	a	b	a	b
Name	L1	C1	L2	C2	L3	C3	L4
Upper Bound	40	400	300	300	50	1500	100
Lower Bound	10	100	100	100	10	500	10
Optimized	36	337	182	184	15	1429	28
Type	a	b	b	b	a	b	a
Name	C4	L5	L6	L7	C5	L8	C6
Upper Bound	300	200	300	200	300	300	300
Lower Bound	100	50	50	10	50	30	50
Optimized	283	71	221	93	67	178	207
Type	b	a	b	b	a	b	a
Name	L9	C7	L10	L11	C8	L12	C9
Upper Bound	200	300	100	200	300	500	200
Lower Bound	50	50	30	50	100	100	100
Optimized	129	139	51	128	168	334	110
Type	b	a	b	c	c	c	
Name	L13	C10	L14	W1	W2	W3	
Upper Bound	500	200	200	100	100	100	
Lower Bound	100	50	100	40	40	40	
Optimized	171	82	152	60	80	60	

As shown in Figure 4.3, the passive components, such as capacitors and microstrip lines in the input, output, and intermediate matching circuits, are connected and grouped into four parameterized sub-circuits (i.e., matching circuits 1-4 in dashed circles). These sub-circuits are interconnected at the top-level circuit for holistic characterization. The

drain voltage and gate voltage of all stages are fixed at 12 V and -1.25 V, respectively. There are 27 design variables in total, as marked in Figure 4.4. These variables can be categorized into three types: (a) values of capacitors (fF), (b) widths and lengths of microstrip lines (μm), and (c) the gate widths of transistors (μm). Table 4.3 lists all design variables along with their search ranges (lower and upper bounds).

Table 4.4: Design specifications (Example 1)

Types	**Items**	**Band**	**Specifications**
Specification 1	Input Matching / S_{11}	27-31 GHz	$\leq$ -11 dB
Specification 2	Output Matching / S_{22}	27-31 GHz	$\leq$ -11 dB
Specification 3	Gain / S_{21}	27-31 GHz	$\geq$ 18 dB
Specification 4	Gain Ripple / ΔS_{21}	27-31 GHz	$\leq$ 1 dB
Specification 5	PAE	27-31 GHz	$\geq$ 22 %
Specification 6	Output Power	27-31 GHz	$\geq$ 34 dBm
Specification 7	AMPM	27-31 GHz	$\leq$ 20 deg

The design specifications of this example are provided in Table 4.4. Among these specifications, input matching, output matching, and gain are characterized by S-parameter, specifically, i.e., S_{11}, S_{22}, and S_{21}. Gain ripple refers to the variations in the gain across the operating band. Output power denotes the RF power delivered from PA to the load. PAE refers to power-added efficiency, defined by

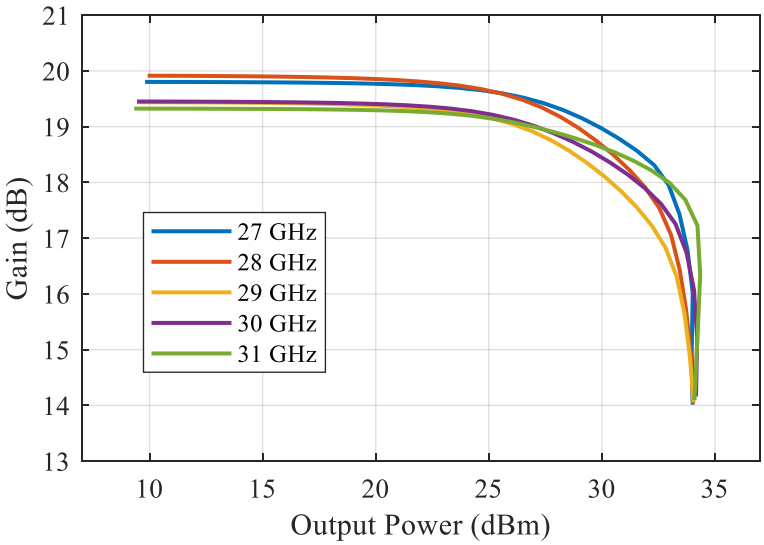
$$\text{PAE} = \frac{P_{\text{out}} - P_{\text{in}}}{P_{\text{DC}}} \times 100\% \tag{4.2}$$

where P_{out} is the RF output power, P_{in} is the RF input power, and P_{DC} is the DC power supplied to the PA. AMPM refers to amplitude-to-phase modulation, which describes the linearity of PA by measuring the change in phase of the output signal as the amplitude of the input signal varies.

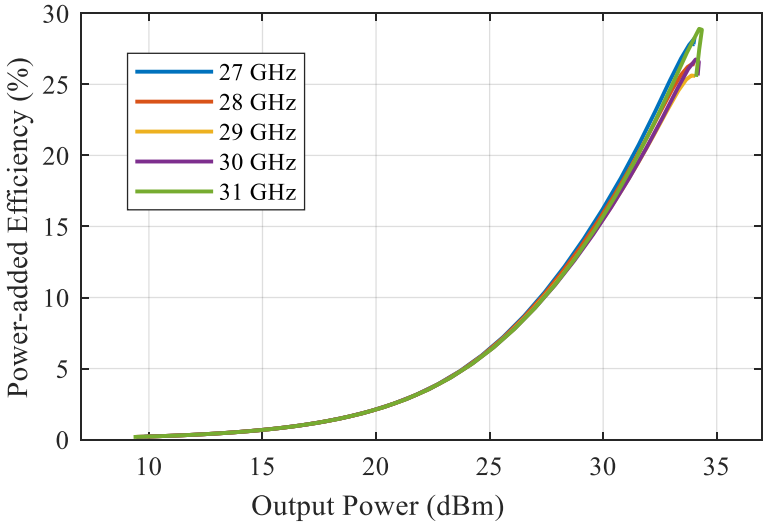
Table 4.5: Performance of a typical optimal design by proposed method (Example 1)

Item	**Band**	**Worst In-band Value**
Input Matching / S_{11}	27-31 GHz	-10.5 dB
Output Matching / S_{22}	27-31 GHz	-12.47 dB
Gain / S_{21}	27-31 GHz	19.32 dB
Gain Ripple / ΔS_{21}	27-31 GHz	0.66 dB
PAE	27-31 GHz	25.62 %
Output Power	27-31 GHz	34 dBm
AMPM	27-31 GHz	15.8 deg

For each specification, the worst value across the given band is used as the performance metric. Due to the pre-designated RC tanks before the input port of each stage, the stability factor is not included as a specification here; however, it is monitored, and no violations were observed during the optimization process. It can be seen that this example takes most small- and large-signal PA metrics into full consideration than works in [134, 135, 146].



(a) AMAM performance



(b) PAE performance

Figure 4.5: The performance of a typical optimal design, Example 1

In terms of simulation time, each simulation takes approximately 5-6 minutes and the maximum simulation budget is set to 1200 (about five days). Note that, this budget was deliberately overestimated to ensure convergence and allow for a fair comparison between two algorithms. The following result quickly illustrate this point: In the four independent runs using the proposed algorithm, all of them successfully satisfy all the

specifications, with the objective function reaching zero after 551, 465, 545, and 503 simulations, respectively. The average value across these runs is approximately 516 simulations (52 hours). A typical optimal design is shown in Table 4.3. Its large-signal performances, including gain and power-added efficiency versus output power within the operation band, are shown in Figure 4.5. The performance metrics are summarized in Table 4.5. It can be observed that for this challenging PA design problem, the proposed algorithm can obtain layout-level high-performance designs within about 2 days. Compared to the manual design process that took several weeks [153], efficiency has been greatly improved. Note that, ADS built-in optimization tools are not able to find feasible solutions within the given simulation budget.

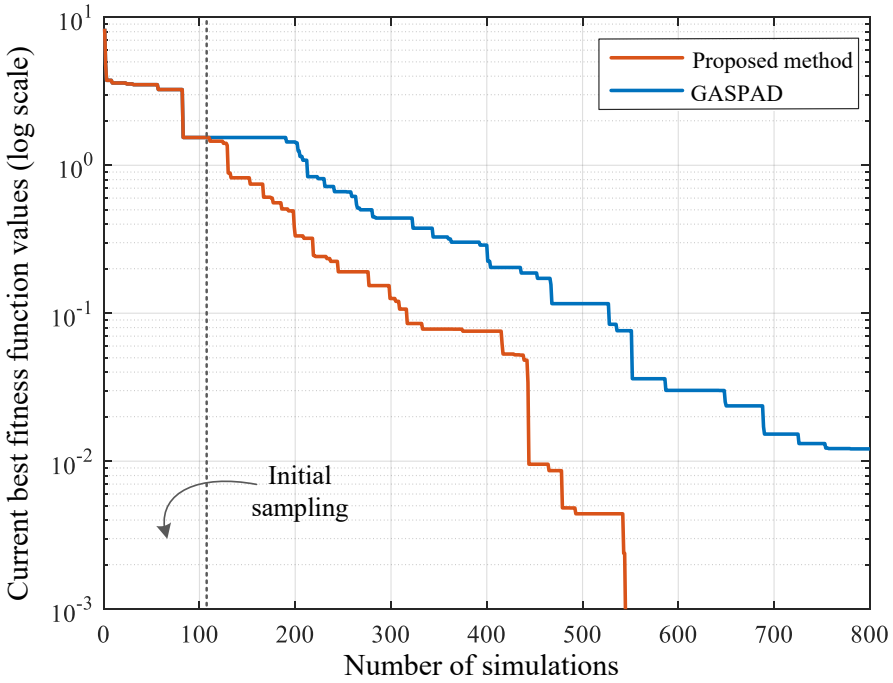


Figure 4.6: The convergence trends of the proposed method and GASAPD (average of four runs, Example 1)

In contrast, in the four independent runs of GASPAD using the same initial populations, only two of them successfully satisfy all the specifications within 1200 simulations (using 1186 and 1097 simulations, respectively). For the other two runs, the best designs achieved were still far from satisfying the specifications. The average convergence trend is shown in Figure 4.6, demonstrating a significant superiority in both efficiency and effectiveness.

4.5.2 Example 2: A 24-31 GHz Wideband Doherty MMIC PA

The second example is a wideband Doherty PA with two stages: a driver stage and a final stage. The input signal is split by a coupler with an isolation resistor, and then fed forward into two branches with different bias classes (i.e., the main branch and the auxiliary branch). The two branches amplify the signal separately and are combined at the output without isolation. The design of a Doherty PA is difficult due to an active load-pull interaction between the main and auxiliary branches and the need to ensure proper phase and amplitude alignment. The complexity is further increased for a multistage Doherty PA that operates over a wide bandwidth, as is the case for this example.

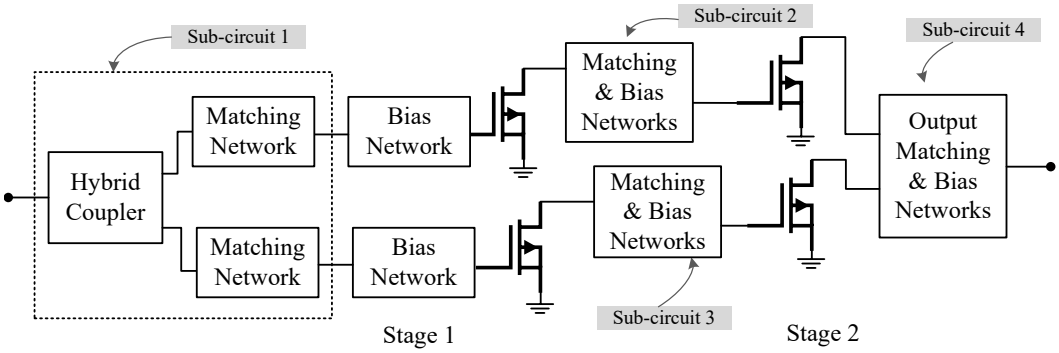


Figure 4.7: Top-level schematic of the wideband Doherty MMIC PA.

As shown in Figure 4.7, the passive components are connected to form four sub-circuits: an input coupler with matching circuits for the first stage (sub-circuit 1), two intermediate matching circuits (sub-circuits 2 & 3), and an output combining and matching circuit (sub-circuit 4). The top-level schematic is depicted in Figure 4.7. In each sub-circuit, the microstrip line is pre-folded into a specific shape to generally satisfy the space constraints. The details of this rule-based prefolding approach are illustrated in Figure 4.8. Subsequently, design variables are assigned to each segment to maintain the shape of the folded line. Note that prefolding rules are defined manually by the designer and are not automated by the algorithm.

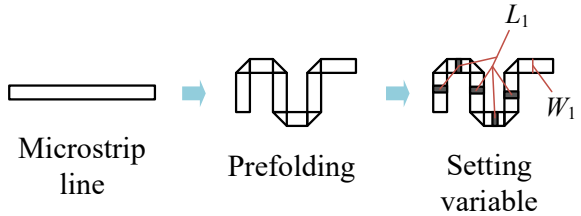


Figure 4.8: Illustration of microstrip line prefolding.

Table 4.6: Design variables, search ranges, and a typical optimal design obtained by proposed method (Example 2)

Type	b	b	a	a	a	c	b	b
Name	L1	L2	C1	C2	C3	R1	L3	L4
Upper Bound	200	550	200	200	200	100	50	50
Lower Bound	50	400	100	100	100	40	-30	-30
Optimized	112	539	130	183	103	53	-25	9
Type	a	b	b	b	b	a	a	b
Name	C4	W1	L5	L6	L7	C5	C6	W2
Upper Bound	200	70	50	50	50	200	200	70
Lower Bound	100	50	-30	-30	-30	100	100	50
Optimized	188	64	-16	-26	-19	164	199	51
Type	b	b	b	b	b	a	a	a
Name	L8	W3	L9	L10	L11	C7	C8	C9
Upper Bound	250	70	250	250	150	200	300	300
Lower Bound	100	50	100	100	50	80	200	200
Optimized	196	57	141	160	85	137	205	262
Type	a	a	a	b	b	b	b	
Name	C10	C11	C12	W4	L12	L13	L14	
Upper Bound	200	150	1400	85	300	50	50	
Lower Bound	100	50	800	65	100	-50	-50	
Optimized	124	53	1281	79	279	-38	-46	

There are 31 design variables in total, which are marked in Figure 4.9. These variables are categorized into four types: (a) capacitor values (fF), (b) widths and lengths of microstrip lines (μm), and (c) resistor values (Ω). The transistor parameters, such as the number of fingers and gate width, are fixed by the designer in this example at 4 and 80 μm in the driver stage and 6 and 80 μm in the final stage, respectively. Table 4.6 lists all of the design variables along with their search ranges. The drain voltage of all stages is 12 V; the gate voltage is -1.25 V for the main branch, and -2.6 V and -2.2 V for the auxiliary branches, respectively.

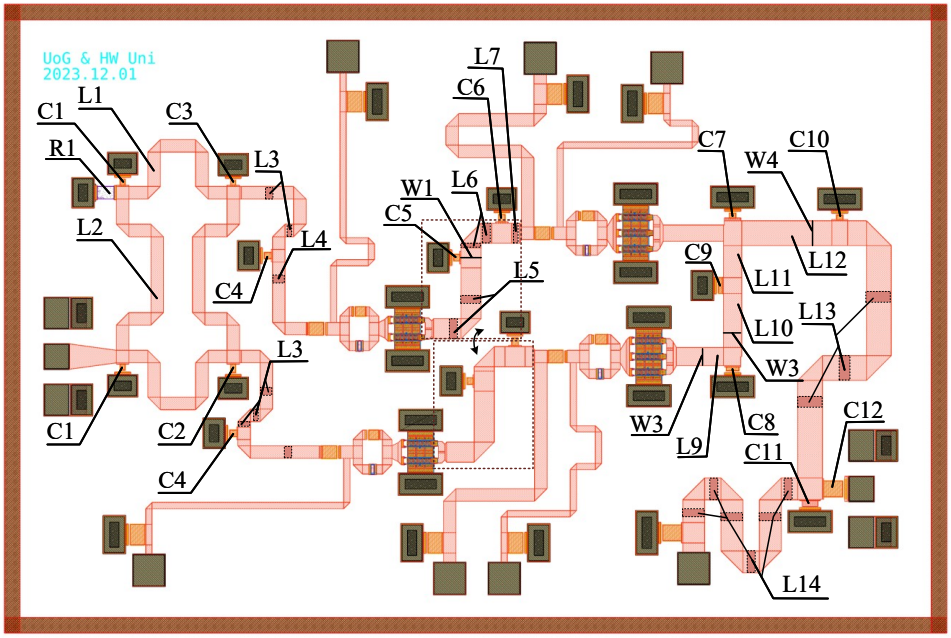
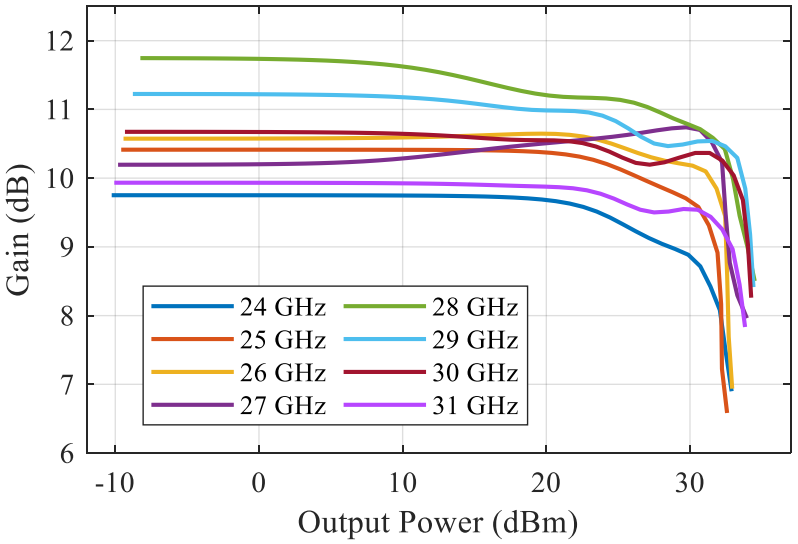


Figure 4.9: The layout photo of the wideband Doherty MMIC PA

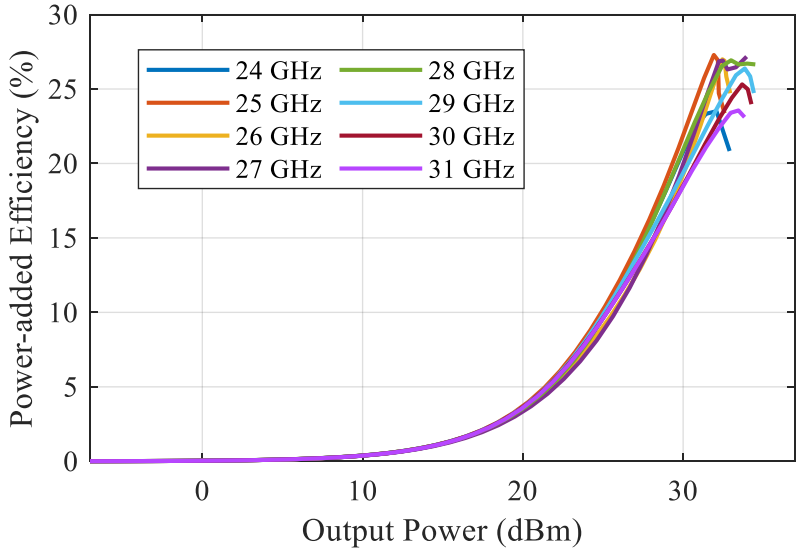
Table 4.7: Design specifications (Example 2)

Type	**Item**	**Band**	**Specification**
Specification 1	Input Matching / S_{11}	24-31 GHz	$\leq$ -7 dB
Specification 2	Output Matching / S_{22}	24-31 GHz	$\leq$ -7 dB
Specification 3	Gain / S_{21}	24-31 GHz	$\geq$ 8.8 dB
Specification 4	Gain Ripple / ΔS_{21}	24-31 GHz	$\leq$ 2.5 dB
Specification 5	PAE	24-31 GHz	$\geq$ 20 %
Specification 6	6dB backoff PAE	24-31 GHz	$\geq$ 14 %
Specification 7	Output Power	24-31 GHz	$\geq$ 34 dBm
Specification 8	AMPM	24-31 GHz	$\leq$ 20 deg
Specification 9	Maximum Main Pout	24-31 GHz	$\geq$ 31 dBm
Specification 10	Maximum Peak Pout	24-31 GHz	$\geq$ 31 dBm

The specifications of this wideband Doherty PA are shown in Table 4.7. Specifications 1 to 8 are large- and small-signal performances as the same in Example 1. Note that the 6^{th} specification is defined specifically to Doherty PA with PAE at 6 dB backoff being used. The last two specifications, which concern the maximum output power at the main and auxiliary branches, are included to prevent the optimization from resulting in undesirable performance (e.g., where almost all the output power comes only from the main branch).



(a) AMAM performance



(b) PAE performance

Figure 4.10: The performance of a typical optimal design, Example 2

For this PA, each simulation takes about 6 to 7 minutes and the maximum budget is 1000 simulations (i.e., about five days). In the four independent runs of the proposed method, all successfully satisfy the specifications, with the objective function reaching zero after 469, 713, 603, and 513 simulations, respectively. The average across these runs is approximately 574 simulations (60 hours). A typical optimal design is shown also in Table 4.6. Its large-signal performances, gain, and power-added efficiency versus output power across the operation band are shown in Figure 4.10. All performance metrics are summarized in Table 4.8. It can be observed that considering the 7 GHz bandwidth requirement, the design obtained by the proposed method is promising within just about 2.5 days. As for comparison, GASPAD, exhausting the simulation budget, and the best designs obtained in the four runs were far from satisfactory. The average

convergence trend shown in Figure 4.11 also demonstrates significant superiority in both efficiency and effectiveness, similar to the results observed in Example 1. Still, ADS built-in optimization tools are not able to find feasible solutions within the given simulation budget.

Table 4.8: Performance of a typical optimal design by proposed method (Example 2)

Item	**Band**	**Worst In-band Value**
Input Matching / S_{11}	24-31 GHz	-7.05 dB
Output Matching / S_{22}	24-31 GHz	-9.7 dB
Gain / S_{21}	24-31 GHz	9.7 dB
Gain Ripple / ΔS_{21}	24-31 GHz	1.8 dB
PAE	24-31 GHz	20.2 %
PAE at backoff	24-31 GHz	14.9 %
Output Power	24-31 GHz	34.0 dBm
AMPM	24-31 GHz	10.8 deg
Maximum Main Pout	24-31 GHz	32.53 dBm
Maximum Peak Pout	24-31 GHz	31.55 dBm

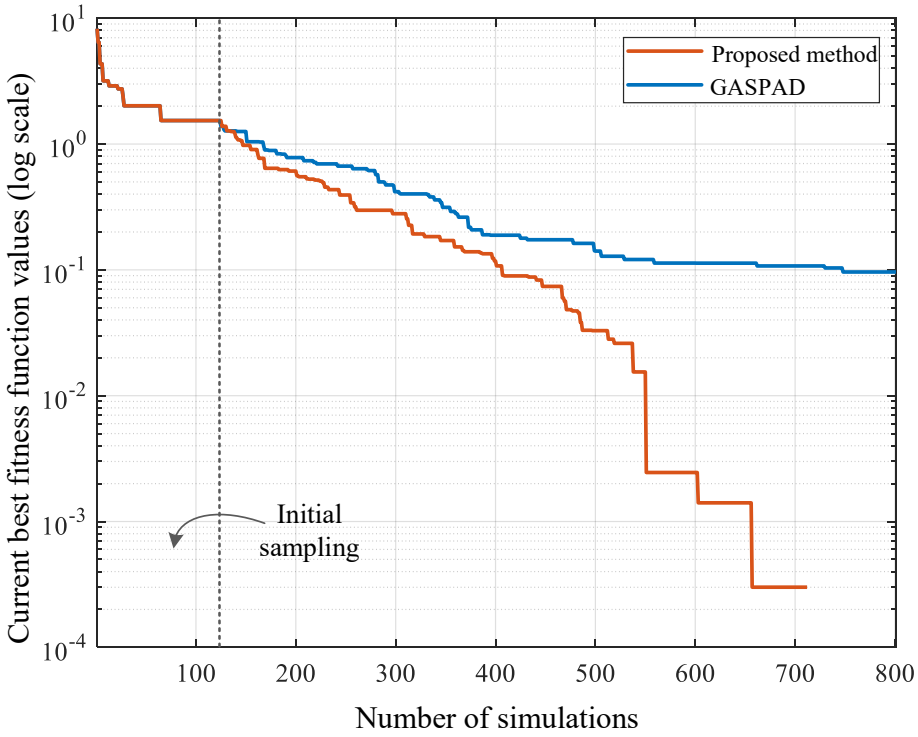


Figure 4.11: The convergence trends of the proposed method and GASAPD (average of four runs, Example 2)

Table 4.9 provides a comparison of performance figures of merit with several Ka-band Doherty PAs published within the last 5 years. It is demonstrated that our optimized design achieves the broadest bandwidth while maintaining comparable performance in other aspects. Although this comparison is not very rigorous, as our results are only from simulations, it still serves to demonstrate the effectiveness of the proposed methodology.

4.6 Summary

In this chapter, the design automation for MMIC PA at the layout level is explored. The chapter begins with an introduction to the conventional PA design methodology and its challenges, followed by a thorough review of techniques proposed over recent decades for PA design automation. The primary problem addressed in this research is then identified and discussed in detail. The proposed methodology consists of two key components: the optimization-oriented integrated environment and the BNN-based optimization algorithm, which together enable a higher degree of MMIC PA design automation. The optimization-oriented integrated environment, bridging the external programming language with ADS, is compatible with most PDKs and current design workflows and serves for holistic circuit characterization. To the best of my knowledge, it is proposed for the first time. The main innovation of the proposed algorithm lies in the use of BNN-based prediction and prescreening during optimization, combined with the hybrid search engine introduced in SAEA frameworks. The effectiveness of the proposed methodology is validated by two MMIC PAs: a 27-31 GHz balanced PA and a 24-31 GHz wideband Doherty PA, with the latter having been taped out for manufacturing. The average number of simulations required for both PAs is approximately 500, demonstrating good efficiency and outperforming most of the published work discussed in the literature review.

Ref.	**Year**	**Frequency (GHz)**	**FBW (%)**	**Technlogy**	**Stages**	P_{out}**@Sat. (dBm)**	**Small-signal Gain (dB)**	PAE**@Sat. (%)**	PAE**@6dB-backoff (%)**
[154]	2019	28	-	GaN-Si	two	32	13	20	30
[155]	2021	24–28	15.4	GaN-SiC	two	34–36	18–19	23–41	14–32
[156]	2022	16.3–20.3	22	GaN-Si	three	36.6–37.7	26–29**	23–31	19–21
[157]	2022	24–28	15.4	GaN-SiC	two	35.4–36	14.9–19.7	27.8–36.8	18.1–30.1
[158]	2023	28–29	3.5	GaN-SiC	three	34–34.3	15–20	20–22	13–16
[159]	2024	25.5–27	5.7	GaN-SiC	one	31.4–32.1	4.8–7.2**	30–37.2	27–30.5
Ours*	-	24–31	25.45	GaN-Si	two	34.1–35.1	9.7–11.4	20.2–27.2	15–18.8

Table 4.9: Comparison of performance figures of merit with published work

* : The simulation results.
** : The figures are read from graphs.

Chapter 5

Algorithmic Design Optimization for Semiconductor Devices

5.1 Background

Semiconductor devices (simplified as devices hereinafter) are critical components in modern electronic systems because of their ability to control and manipulate electrical signals. Among the different categories of devices, transistors are a key type and serve as the fundamental building blocks for most electronic circuits [160]. However, the design of transistors has been somewhat artisanal till now. For most of the history of semiconductor technology, the industry relied mainly on shrinking transistor geometries for performance improvements [161]. New devices are developed, designed, fabricated, and validated in laboratories using a trail-and-error method [17]. Due to the complexity of physical processes in fabrication as well as their variations, simulation based on quantitative or semi-quantitative equations is often considered inaccurate or even intractable, with experimental designing and testing being the gold standard of validation.

However, the use of new technology and materials has significantly increased the complexity of transistor structures, requiring extensive experiments due to the numerous combination options in the design process, which greatly increases time-to-market [162, 163]. This complexity has driven the development of technology computer-aided design (TCAD), enabling the simulation of processes and devices to predict performance to some extent in advance [16]. Despite these achievements, there are often delays between qualitative understanding and quantitative reproduction within TCAD and the actual application of a technology node [164]. Consequently, in the semiconductor domain,

optimization generally refers to improvements based on theory and experience rather than algorithmic pathway [165]. The advantage of this methodology is that this *optimization* outcomes are often accountable and interpretable, while the downside is that the results are likely to be suboptimal.

Table 5.1: Publications on DTCO over past 15 years (1994-2021)

Ref.	Article type	Year	Algorithm involved?	Design optimized?	Device/Circuit
[166]	Journal	2021	YES	NO	Compact Model
[167]	Conference	2020	NO	YES	CMOS
[168]	Conference	2020	NO	YES	CMOS
[169]	Conference	2020	NO	YES	FinFET
[170]	Conference	2020	NO	NO	CMOS
[171]	Conference	2020	NO	NO	DRAM
[172]	Journal	2020	YES (SA)	YES	System
[173]	Journal	2020	NO	NO	Ga_2O_3 SBD
[174]	Conference	2020	YES	NO	MOSFET
[175]	Journal	2019	NO	NO	FinFET
[176]	Journal	2019	YES	YES	Compact Model
[177]	Journal	2018	NO	YES	Carbon Nanotube
[178]	Conference	2018	NO	YES	FinFET
[179]	Conference	2017	NO	NO	FinFET
[180]	Conference	2017	YES (PSO)	YES	CMOS
[181]	Journal	2015	NO	NO	FinFET
[182]	Journal	2015	YES (GD)	YES	CNFET
[183]	Journal	2015	NO	NO	CMOS
[184]	Conference	2014	No	NO	CMOS
[185]	Journal	2012	NO	Yes	CMOS
[186]	Journal	2010	NO	YES	CMOS
[187]	Conference	2009	NO	NO	MOSFET
[188]	Journal	2008	NO	YES	CMOS
[189]	Journal	2007	YES	NO	FPGA
[190]	Conference	2007	NO	NO	CMOS
[191]	Journal	2005	NO	YES	CMOS
[192]	Journal	2004	YES	NO	NMOS
[193]	Journal	1999	NO	YES	Bipolar
[194]	Conference	1994	NO	Yes	MOSFET

A critical methodology that must be mentioned is the concept of design technology co-optimization (DTCO) proposed for silicon technology [195] around twenty years ago. In DTCO, due to the complexity of advanced nodes (e.g., FinFET), the manufacturing process and device design are optimized simultaneously for better performance and lower yield loss. Table 5.1 lists publications on DTCO from 1994 to 2021 (the time when this research was conducted). At least three conclusions can be drawn. First, in most publications, algorithms are not involved, even though the device or circuit is truly optimized by manual trail-and-error. Second, even when algorithms are involved, they are always off-the-shelf global or local optimization algorithms, such as simulated annealing, particle swarm optimization, and gradient descent, lacking specialized algorithms designed for device optimization. Third, the concept of DTCO primarily targets silicon devices, with very little or no work on other technologies like III/V semiconductors.

Therefore, in this chapter, an attempt of algorithmic design optimization for semiconductor devices is presented, showcased by two case studies: a terahertz InP pHEMT design optimization and a novel concept of device circuit co-optimization for CMOS. For design optimization of semiconductor devices, efficiency (measured by iterations) is the most crucial factor due to their high computational cost, which often exceeds that of electromagnetic simulations. Hence, the optimization algorithms used and proposed in these two studies are carefully devised to suit the scenario. Efficiency and effectiveness are validated through outcomes.

In the following section, the implementation of TCAD interface is first introduced, forming the foundation for further research. Then two case studies are presented sequentially, followed by a conclusion in the summary section.

5.2 Preliminary: Implementation of TCAD Interface

To enable algorithmic design optimization in TCAD, simulation software such as Silvaco or Sentaurus must be controlled and interfaced with external programming languages. However, off-the-shelf tools that facilitate this process are often unavailable. Therefore, it is necessary to implement a custom TCAD interface as a preliminary step. This section provides a detailed explanation of the TCAD interface, with the specific focus on Sentaurus, as the following case study is based on this software.

Understanding the simulation process and file structures is essential before interacting with TCAD software. In Sentaurus, a simulation project consists of several model (physical structure) files and command files that combine different numerical solvers. The entire simulation process is organized through a tree file called `gtree.dat`, which sequentially populates design variables into different nodes. An example of `gtree.dat` is provided in Appendix C. When the simulation is launched, the `gtree.dat` file scans these nodes and locates the necessary command files. Once the simulation is complete, results can be exported to a `.csv` file by commands linked to specific nodes.

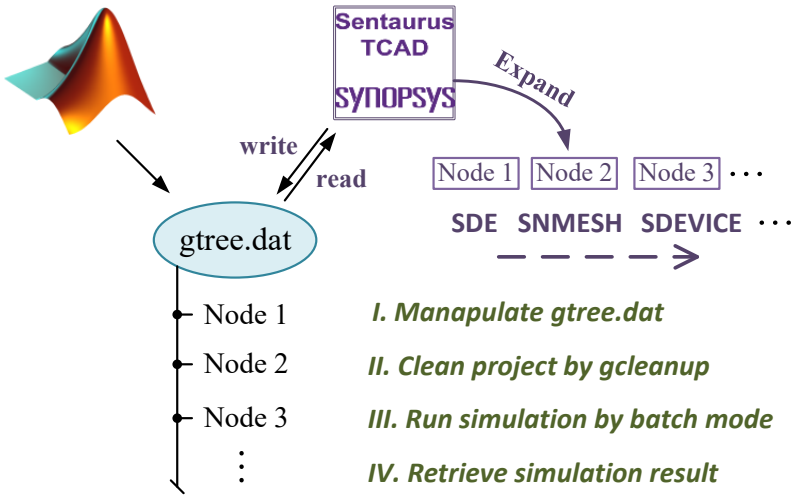


Figure 5.1: Illustration of procedures of TCAD interface.

Note that Sentaurus offers a batch mode that allows simulations to run directly via the command line, provided the `gtree.dat` file is correctly organized. This enables the implementation of an interface. Figure 5.1 summarizes the main procedures of the Sentaurus interface and the aforementioned simulation process. An external programming language, in this case, MATLAB is used to manipulate the `gtree.dat` file by populating new variable values, which are then read by the Sentaurus main program. As the tree file expands, different nodes correspond to different solvers, such as SDE, SNMESH, and SDEVICE. Before launching a new simulation, the project is cleaned and refreshed, followed by running the simulation in batch mode and retrieving the results.

It is important to note that this pipeline is not limited to Sentaurus or specific TCAD software. It can be widely applied to other simulation software that supports command-line execution. An example code is provided also in Appendix C.

5.3 Case Study 1: Terahertz pHEMT Design Optimization

5.3.1 Introduction

Terahertz (THz) frequency, ranging from 100 GHz to 10 THz [196], has gained much attention in various applications, such as security screening [197], next-generation autonomous radars [198], and high data rate mobile communications beyond 5G. Transistors, the core devices of RF system front-ends, benefit from indium phosphide (InP)-based pseudomorphic high electron mobility transistors (pHEMTs), which exhibit a cut-off frequency f_{T} (i.e., the frequency at which the transistor current gain drops to unity [199]) over 700 GHz and a maximum oscillation frequency $f_{\max}$ (i.e., the frequency at which the power gain drops to 0 dB) nearing 1.5 THz [200]. These characteristics make them ideal for terahertz monolithic integrated circuits, offering advantages in cost, integration, and compactness over other technologies.

However, designing pHEMTs is often not trivial, requiring optimization of both materials of epitaxial layers and structures of electrodes, with over 20 parameters involved. For example, To achieve the highest operating frequency, one must first optimize materials—such as material composition, layer thickness, and doping concentration—to increase electron mobility as much as possible, followed by structural optimization to reduce parasitic capacitance, including gate length, gate shape, source-drain separation, the gate to drain/source distance, and so on. In addition, gate and source/drain contact materials must also be refined. Traditionally, the above optimization process relied on trial-and-error experiments in a clean room, a time-consuming and costly task requiring hundreds of iterations. Thanks to the extensive development of TCAD, numerical model simulation is now available for the characterization of most electronic devices. However, the intensive computational cost of TCAD simulation hinders the algorithmic device design by directly optimizing key parameters. Therefore, low efficiency due to the trial-and-error methodology still exists. This calls for the need to employ modern optimization techniques.

This section introduces a machine learning-assisted global optimization method for pHEMT design optimization for the first time. A GP-based SAEA algorithm [201] is employed to optimize the structure of a commercial pHEMT [202] and enhance its performance towards terahertz. Depending on different applications, the optimization

focus of pHEMTs varies. For terahertz applications, f_{T} and f_{max} are the main figures of merit and should be maximized wherever possible. Therefore, they are set as the objective of this research. This new method demonstrates great potential in terms of efficiency and optimization quality in ultrafast transistor design and can be extended to other advanced semiconductor devices and circuits. The next subsection provides an overview of the commercial pHEMT structure, followed by details of the optimization method. The results and discussion are concluded in the final section.

5.3.2 Structure and Design Methodology

5.3.2.1 pHEMT Structure

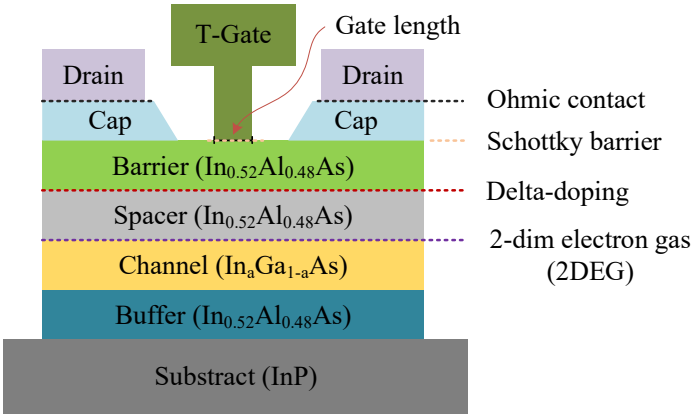


Figure 5.2: Illustration of the structure of pHEMT epilayers and electrodes

Figure 5.2 shows the typical structure of an InP pHEMT. The epilayers consist of several layers, each with different functions, including the InP ground substrate, buffer, channel, spacer, and barrier layers. The materials of each layer used are indicated in brackets, with the barrier, spacer, and buffer layers all composed of $\mathrm{In}_{0.52}\mathrm{Al}_{0.48}\mathrm{As}$. The channel layer is made of $\mathrm{In}_{a}\mathrm{Ga}_{1-a}\mathrm{As}$ to form lattice mismatch, where the indium mole fraction a is adjustable for improved electron mobility [203, 204]. A two-dimensional electron gas (2DEG) is thus formed between the spacer and channel layer [205], confining electrons in a thin layer that can only move freely in the plane of the layer, known as pseudomorphic. To provide more electrons to the channel, silicon delta-doping is introduced between the spacer and barrier, which further enhances the electrical properties regarding electron mobility. Above the barrier layer, a T-shaped Schottky gate

(T-gate) is formed by the contact between metal and semiconductor. The T-gate is used in pHEMT to balance current capacity and parasitic capacitance for higher operating frequency. In addition, the source and drain electrodes form ohmic contacts with the cap layer, providing a low-resistance path for current flow.

5.3.2.2 Traditional Design Method

The epilayers play a significant role in determining the performance of a pHEMT. To improve operating frequency (i.e., f_{T} and $f_{\max}$), both material and structure should be carefully designed. As part of the traditional design methodology, Equation (5.1) and (5.2) express the relationship between the equivalent components and f_{T} and $f_{\max}$, derived from small-signal equivalent circuit [206]. The variables in the right-band term include capacitance (source-gate C_{gs} and drain-gate C_{ds}), resistance (parasitic source R_s, parasitic drain R_d, parasitic gate R_g, output R_{ds}, and channel intrinsic R_i), and transconductance g_m. Therefore, by optimizing the thickness, mole fractions, and material compositions in the barrier, spacer, and channel, these variables can be adjusted to achieve the desired performance. Obviously, this is often a trail-and-error process that heavily relies on experience.

$$f_{\mathrm{T}} = \frac{g_m}{2\pi\left(C_{gs}+C_{gd}\right)\left(1+\left(R_s+R_d\right)/R_{ds}\right)+C_{gd}g_m\left(R_s+R_d\right)} \tag{5.1}$$

$$f_{\max} = \frac{f_{\mathrm{T}}}{2\sqrt{\left(R_g+R_i+R_s\right)/R_{ds}+2\pi f_{\mathrm{T}}R_gC_{gd}}} \tag{5.2}$$

Another straightforward approach to increase f_{T} and $f_{\max}$ is to reduce the gate length. Generally, shorter gate lengths result in better frequency performance. However, when the gate length is reduced below 50 nm, the mechanical support provided by the gate foot becomes insufficient, which can compromise the structural stability and lead to yield losses. Thus, reducing gate length is not always practical, though a successful example has been reported [207] with the gate length around 10 nm. In this work, the gate length was fixed at 100 nm, which is the same as the commercial pHEMT. This aims to verify the capability of algorithmic design optimization in a stringent scenario.

5.3.2.3 Algorithmic pHEMT Design Method

Although the equivalent circuit-based method provides a fundamental approach for extending operating frequencies, it is often implicit and inconvenient, lacking a clear inverse relationship from the values of lumped parameters to the actual pHEMT structures. Additionally, the representation capability of lumped components is limited and typically varies with different bias conditions. As a result, this approach is highly constrained, and the outcomes are often suboptimal.

An alternative approach involves using a global optimizer to search for the optimal pHEMT structure through TCAD simulation. However, given the significant time required for TCAD simulations based on finite element analysis, standard global optimization methods, such as genetic algorithms, often demand thousands to tens of thousands of simulations, making this direction generally impractical.

To reduce the optimization time to a practical level while maintaining the quality of standard global optimization algorithms, SAEAs are introduced into pHEMT design in this work. In SAEA, as introduced in Section 2.3, a surrogate model mapping the inputs (i.e., design variables) to outputs (i.e., performances) is constructed using machine learning techniques. By replacing the computationally expensive Sentaurus simulations with computationally cheaper surrogate model predictions, the optimization time can be considerably reduced. Specifically, the algorithm is implemented in MATLAB, where parameters are passed to Sentaurus for simulation through the aforementioned interface, and the resulting f_{T} and $f_{\max}$ values are returned upon completion of the simulation. Notably, this is the first attempt to apply a machine learning-assisted algorithm to the design of pHEMTs.

Regarding the details of the optimization algorithm, the framework is inherited from Algorithm 6 due to its effectiveness in handling expensive optimization tasks, with modifications to some operators for better adaptation to pHEMT design problems. Latin hypercube sampling is chosen as the initial sampling method to uniformly initialize the design space. In each iteration, all simulated designs are ranked in descending order by $f_{\mathrm{T}} + f_{\max}$, and the top 100 designs are selected to form the parent population. The offspring population is then generated by applying DE mutation and crossover operators. To maximize efficiency, the DE/best/1 mutation strategy is employed for its fast convergence. The construction of GP surrogate models follows the same manner

as in Algorithm 6. The LCB prescreening method introduced in Section 2.3.1 is also used to account for both prediction uncertainty and performance, identifying the most promising design to simulate with Sentaurus. This process continues until the stopping criterion is met.

5.3.3 Result and Discussion

To optimize the design of the pHEMT, the device must first be modeled and calibrated in TCAD environment to match real-world performance as documented in the datasheet [202]. The TCAD models used in Sentaurus include the hydrodynamic transport model for electrons, high-field mobility, and recombination models such as Shockley–Read–Hall (SRH), Auger, and Radiative recombination. Both ohmic and Schottky contact are defined. The properties of the materials used in the simulation are listed in Table 5.2, with the indium mole fraction a set to 0.53. All simulations are conducted under room temperature conditions. The pHEMT structure is modeled in two dimensions, as illustrated in Figure 5.2. To ensure simulation accuracy, the mesh density below the gate region is increased. The calibrated results, i.e., the transfer characteristics, are shown in Figure 5.3, and performance metrics are summarized in Table 5.4, showing a good alignment between the simulation and the results of the datasheet.

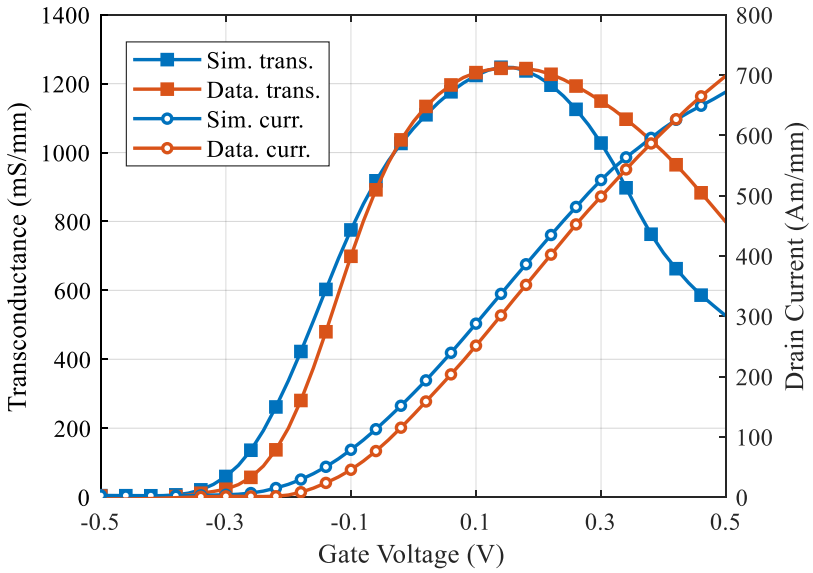


Figure 5.3: Comparison of transfer characteristics between calibrated simulation and datasheet results.

Table 5.2: Semiconductor parameters used in the TCAD simulation.

Parameter	**InP**	$\mathbf{In}_a\mathbf{Ga}_{1-a}$	$\mathbf{In}_{0.52}\mathbf{Al}_{0.48}$
Lattice constant (Å)	5.86	5.86	5.86
Band gap (eV)	1.34	0.72	1.48
Dielectric constant (static)	12.4	14.3	12.4
Electron affinity (eV)	4.44	4.55	4.27
Effective mass m_c^*/m_o at central valley	0.079	0.047	0.081

For design optimization, 15 key design variables were selected, covering most tunable parts of the pHEMT structures. The gate length is maintained 100 nm, consistent with the same as the commercial design. Table 5.3 outlines the search range for each variable. Besides these variables, variations in the Schottky barrier and contact resistance are also considered but not listed here. Geometric constraints were applied during optimization to ensure the physical feasibility of the parameters. Thus, the optimization problem can be formulated as follows:

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{argmax}} \quad & (f_{\text{T}},\, f_{\text{max}}) \\ \text{subeject to:} \quad & f_{\text{T}} \geqslant 220\text{GHz} \\ & f_{\text{max}} \geqslant 550\text{GHz} \\ & x_4 + x_5 - x_{10} \geqslant 3\,(\text{nm}) \\ & x_{10} - x_{12} \geqslant 2\,(\text{nm}) \\ & x_{13} - x_{14} \geqslant 0.35\,(\mu\text{m}) \\ & x_{13} + x_{15} \leqslant 1.15\,(\mu\text{m}) \end{aligned} \tag{5.3}$$

The final optimized pHEMT design is shown in the last column of Table 5.3, with transfer characteristics presented in Figure 5.4, and performance metrics summarized in Table 5.4. The Schottky barrier is set to 0.6 eV, and the contact resistance is 30 $\Omega \cdot \mu$m. The results show that f_{T} and f_{max} are improved to 336 GHz and 770 GHz, respectively, compared to the commercial design's 220 GHz and 550 GHz, representing 57% and 37% improvements without altering the gate length. Additionally, the maximum transconductance and drain current are improved from 1255 to 1672 mS/m and from 675 to 775 mA/mm, respectively, even though these were not primary optimization objectives. Overall, the optimized pHEMT achieves higher operating frequencies in terms of f_{T} and f_{max}, as expected. The optimization process took 16 hours on a standard desktop computer, which is significantly faster than traditional methods that typically take several days.

Table 5.3: The search ranges of the design parameters and the optimized value.

Var.	Parameter name	Search range	Opt. value
x_1	The thickness of substrate layer (μm)	50 - 100	61
x_2	The thickness of buffer layer (μm)	0.2 - 0.8	0.2
x_3	The thickness of channel layer (nm)	4 - 20	4
x_4	The thickness of spacer layer (nm)	2 - 10	2
x_5	The thickness of barrier layer (nm)	8 - 15	8
x_6	The thickness of cap layer (nm)	8 - 25	8.5
x_7	Cap bulk concentration (cm^2/Vs)	1e18 - 2e19	1.40e18
x_8	Delta-doping concentration (cm^2/Vs)	1e12 - 1e13	9.40e12
x_9	Indium fraction of channel layer	0.6 - 0.85	0.85
x_{10}	Location of delta doping (nm)	4 - 21	7
x_{11}	Passivation thickness (nm)	40 - 100	40
x_{12}	Recessed thickness (nm)	0 - 7	5
x_{13}	Gate foot location (μm)	0.45 - 1.05	0.47
x_{14}	Gate-source separation (μm)	0.1 - 0.6	0.1
x_{15}	Gate-drain separation (μm)	0.1 - 0.6	0.1

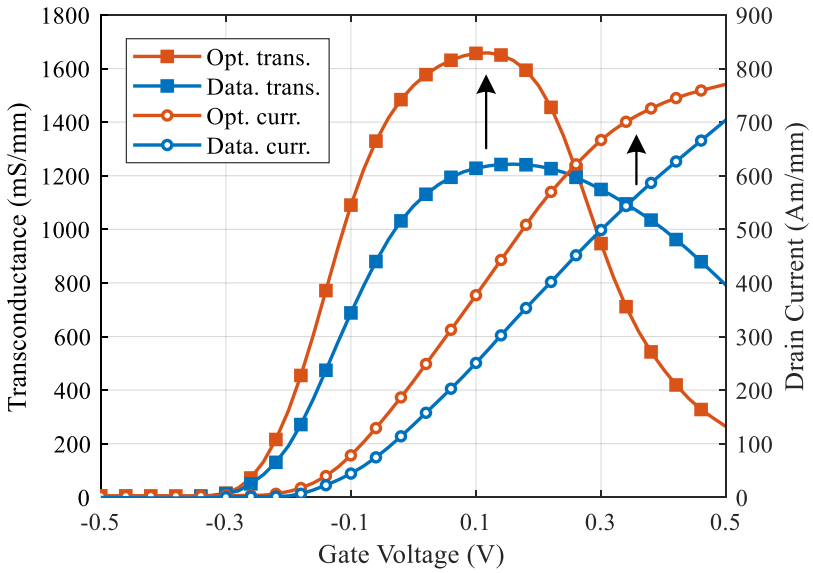


Figure 5.4: Comparison of transfer characteristics between optimized and commercial results.

In conclusion, this work presents a machine learning-assisted design optimization method for pHEMTs. A commercial 100 nm pHEMT was modeled in TCAD simulation, and the structure of its epitaxial layers was optimized for higher cut-off frequency and maximum oscillation frequency. Significant improvements were achieved, with a 57% increase in cut-off frequency and a 30% increase in maximum oscillation frequency com-

Table 5.4: Performance comparison among the commercial, calibrated and optimized pHEMTs

Performance	**Commercial**	**Calibrated**	**Optimized**
Transconductance (mS/mm)	1250	1255	1672
Drain Current (mA/mm)	700	675	775
f_{T} (GHz)	220	215	336
f_{max} (GHz)	550	542	770

pared to the commercial design. The optimization took just 16 hours on a standard desktop computer, using 200 iterations, a substantial reduction in time compared to the traditional trial-and-error method, which usually takes several days. Additionally, the maximum transconductance and drain current were improved to 1600 mS/m and approach 800 mA/mm, respectively. This method demonstrates high potential in terms of efficiency and optimization quality for transistor design regarding different performance metrics and can be seamlessly extended to other advanced semiconductor devices and circuits.

5.4 Case Study 2: Device Circuit Co-Optimization

5.4.1 Introduction

In the current era of semiconductor technology, numerous advancements are taking place at the device level, including the development of FinFET, nanosheet [208], and gate-all-around transistors [8]. These devices empower circuit engineers to design intricate circuits, such as memory and processors, with higher efficiency in terms of reduced power consumption and enhanced processing speed. However, the performance of these circuits relies not only on the precise adjustment of external passive components, such as resistors and capacitors [209], but also, predominantly, on the physical characteristics of the device. These characteristics include the width and length of the device, and also the process and material of the device, such as the doping concentration in the channel region of the CMOS. However, the manual tuning procedure for determining the accurate values of these parameters is a challenging task involving a substantial computing cost on both circuit and technology simulation, which greatly hinders obtaining the optimal performance [210, 211, 212].

Table 5.5: Literature about ML techniques in semiconductor device

Category	Ref.	Year	ML techniques
Device Modeling	[166]	March 2021	ANN
	[213]	May 2023	ANN
	[214]	Nov. 2023	Physics-informed NN
	[215]	Jan. 2024	VAE
	[216]	Jan. 2024	ANN
	[217]	Jan. 2024	Physics-based ANN
	[218]	Jan. 2024	ANN
Device Simulation	[219]	Nov. 2021	ANN, Autoencoder
	[220]	Oct. 2023	Convolutional NN
Process Variation Prediction	[221]	July 2019	ANN
	[222]	Oct. 2019	ANN
	[223]	Apr. 2020	ANN
	[224]	Jan. 2022	ANN
	[225]	Apr. 2023	ANN
Failure Troubleshooting	[226]	Oct. 2019	Linear regression
Device Designing	[227]	Nov. 2021	GP, Active learning
	[228]	Apr. 2022	GP
	[229]	July 2022	ANN
	[230]	Nov. 2023	Decision tree
	[231]	Jan. 2024	Regression
	[232]	Feb. 2024	Knowledge-based ANN

Thanks to the rapid development of machine learning and artificial intelligence [233], many recent studies have demonstrated that methods based on ML can greatly help in many fields of semiconductor industry, including but not limited to device modeling, device simulation, process variation prediction, failure troubleshooting, and device designing. Some of the selected papers with their corresponding ML techniques are listed in Table 5.5. These methods successfully contribute to reducing simulation time through a pre-trained model of electrical characteristics (i.e. I-V relations) for downstream tasks, while effectively obtaining a better design. However, to capture subtle variations of the signal in downstream tasks, the models (either the ML-based model or the conventional compact model) have to be highly accurate, requiring thousands of data, either by simulation or measurement, to be collected for modeling case by case. This makes the modeling phase time-consuming and extends the time-to-market of the

device. Moreover, ML techniques in the reported literature only play an assisting role in helping engineers with performance evaluation, while manual parameter tuning in the circuit level is still vital in the whole design process. Therefore, suboptimal designs are often obtained.

In this section, a novel design methodology that applies the ML technique to co-optimize the device and circuit parameters simultaneously is presented. Driven by directly modeling the performance at the circuit level, the optimization algorithm helps design the device and circuit in an integrated and straightforward way. To the best of our knowledge, this work is the first to utilize the ML technique directly for the entire design workflow of the device and circuit simultaneously. It is worth noting the difference between traditional DTCO and the proposed device and circuit co-optimization. In traditional DTCO, accurate compact models [234] are first constructed and then used for downstream tasks, and this co-optimization is typically achieved by optimizing parameters across several design phases. In contrast, the proposed methodology does not rely on explicit compact models. Instead, circuit performance is characterized and optimized directly. Further details will be explained in the next section.

The CMOS inverter is used as a proof of concept, since the inverter is the fundamental building block in almost any digital circuit. Appropriate design of the inverter greatly facilitates the development of more intricate structures, such as NAND gates, adders, multipliers, and microprocessors, making the process substantially more efficient. The optimization algorithm proposed for this work is called actor-critic optimization, which is a quasi-reinforcement learning algorithm adapted from the deep deterministic policy gradient (DDPG) for efficient optimization. Considering the intensive computational cost of TCAD simulation, efficiency is the most critical factor for the optimization algorithm to be used.

By applying this algorithm, the inverter is synthesized without predefined or nominal configurations. Five types of parameters were considered as the design variables, including gate length, channel width, channel doping concentration for both NMOS and PMOS, area factor, and load capacitance. The goal is to synthesize an inverter from scratch that functionally works well in a given circuit topology and achieves optimal switching performance. It is noted that this work is a proof-of-concept of the feasibility of ML-based optimization techniques for device circuit co-optimization. Promising applications including FinFET, Gate all around (GAA) FET, and nanosheet architecture can be considered in future works.

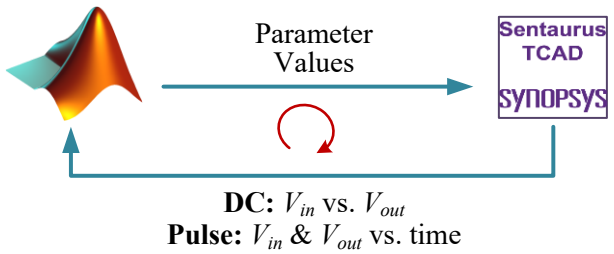
5.4.2 Co-Optimization Methodology

In traditional design workflow, industrial compact models of devices are constructed based on experimental results or thousands of calibrated TCAD simulations [166]. Subsequently, engineers use compact models to carry out circuit-level design, where only the width-to-length ratio of devices can be adjusted, known as transistor sizing. Note that circuit performance is related to both circuit and device parameters, but due to an implicit correlation of the performance with device characteristics, manually tuning these parameters in the current workflow is impracticable.

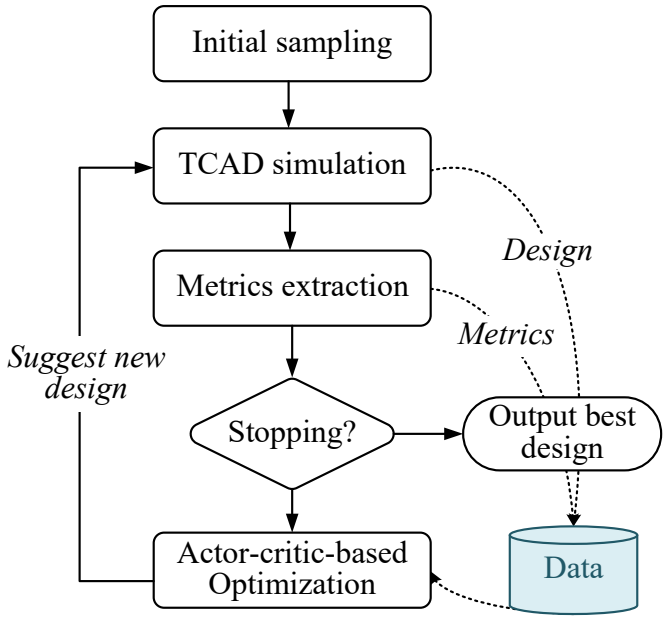
In the proposed methodology, the device and circuit are simulated in a mixed-mode approach (i.e. sequential TCAD and SPICE simulation), while no explicit compact models are constructed. The correlation between device and circuit parameters and performance is learned consecutively through an ML model during optimization. Meanwhile, an algorithm, replacing the manual manipulation of the device parameters (the designer) in the traditional design process, suggests the next candidate solution based on the learned pattern directly.

The dataflow and workflow of our methodology are illustrated in Figure 5.5(a) and (b). In terms of dataflow, the actor-critic-based optimization (i.e., optimizer) is implemented in MATLAB, while the circuit and device simulation models are constructed in Sentaurus (i.e., simulator). For the latter, mixed mode simulation of the circuit and device is performed, including TCAD simulation of the device. In each iteration, the simulator simulates and outputs the performance of a design, and the optimizer provides a new design for the next simulation. This happens iteratively until the optimal design is obtained.

In terms of the workflow in Figure 5.5(b), it commences with several steps. First, a small set of initial sampling for design variables within the predefined search bounds is conducted. For each sample, the circuit performance metrics are extracted from its raw characteristics data. All simulated designs and performance metrics are stored as design-metrics pairs in the dataset. Then the actor-critic-based optimization algorithm is applied to learn the model and suggest the next design with good potential for simulation for the next iteration. Note that only a single new design is suggested for the mixed-mode simulation in each iteration. The iterative process stops when the convergence criteria are satisfied, such as a satisfactory design is obtained, or the computing



(a) Dataflow between MATLAB and Sentaurus



(b) Workflow of the proposed design methodology

Figure 5.5: The dataflow and workflow of the proposed design methodology.

budget (the maximum iterations) is exhausted. Then, the process is terminated and the current best design becomes the output. In the following subsections, the mixed-mode simulation setup, metrics extraction, and optimization algorithm are detailed, respectively.

5.4.2.1 Simulation Setup

Figure 5.6 shows the transfer characteristics (I_{d} vs. V_{g}) of the separate planar NMOS and PMOS devices with two-dimensional architectures, designed using the Sentaurus TCAD structure editor and Sdevice tools [235]. The transfer curves show the expected device behavior for a transistor with a gate length of 50 nm and a height of 10 nm. Here,

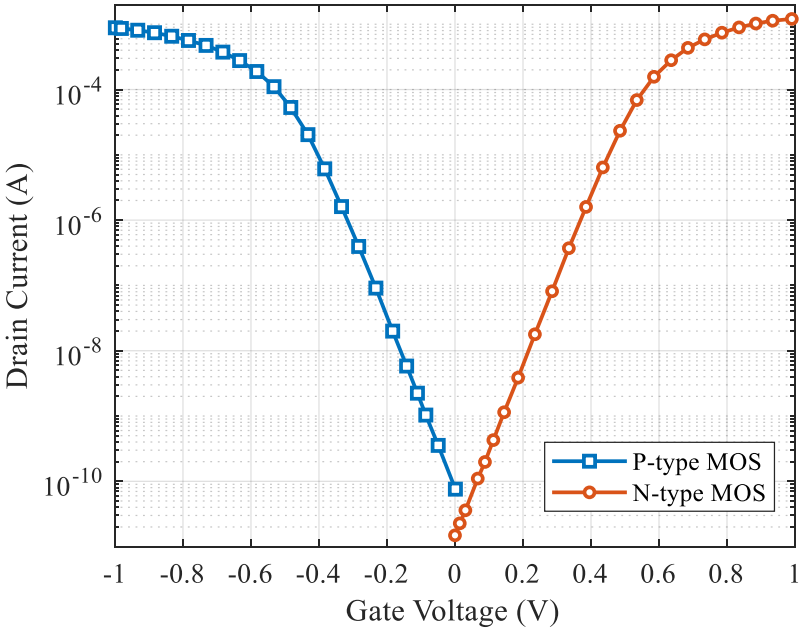


Figure 5.6: Transfer characteristics of the N/P MOSFET

both N- and P-type devices have the same width and channel doping for illustration, and are of enhancement mode. The figure depicts different gate bias conditions required to operate the transistor from liner to saturation regions when $V_{ds} = 0.5$ V. The curves are not strictly symmetric due to the different mobilities of electrons and holes.

Table 5.6: Key parameters in defining the CMOS inverter

Parameter	**Bound / Value**
L_G (Gate length)	14 $\sim$ 90 nm
T_{Si} (Channel thickness)	5 $\sim$ 30 nm
W_{Si} (Aspect ratio of PMOS)	1 $\sim$ 4
* L_{ext} (Source & drain (S&D) length)	30 nm
* EOT (Effective oxide thickness)	2 nm
* N_{SDC} (Doping in the S&D region)	1×10^{18} cm^{-3}
N_{Ch} (Doping in channel for N&PMOS)	$5\times10^{16} \sim 5\times10^{17}$ cm^{-3}
C_L (Load capacitor)	0.01 $\sim$ 10 fF

Table 5.6 displays the complete list of the parameters required to define this device. Parameters that are not set as design variables are marked with (*). In the TCAD simulation, the doping dependence and oldSlotboom mobility along with Shockley-Read-Hall (SRH) recombination models are included. A mixed-mode simulation setup is then developed utilizing an individual MOS transistor. Figure 5.7 shows all the passive components, external node connections, and power supplies needed for the NMOS and PMOS transistors to operate as an inverter circuit.

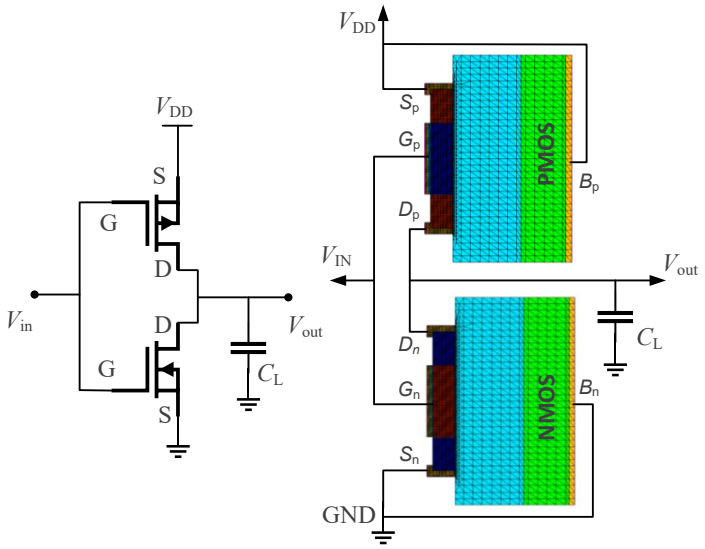


Figure 5.7: Schematics and topology of inverter circuit using NMOS and PMOS device with external node connections and electrical components.

Two types of responses are considered to characterize the performance of the inverter: the voltage transfer response (DC characteristic) and the pulse response (dynamic characteristic). The DC characteristic is depicted by the input versus output voltage of the circuit, denoted by V_{in} and V_{out}, and the dynamic characteristic is depicted by V_{in} and V_{out} over time.

5.4.2.2 Metrics Extraction

For an ideal inverter, the output voltage must trigger, vary, and switch simultaneously as the input pulse rises and falls. This can be manually identified and quantified by defining the figures of merit (FoMs), such as rise time, fall time, edge rate, and propagation delay when the circuit functions effectively [236]. However, when nominal device configurations are not provided, setting these figures as optimization metrics is not straightforward. The red dashed line in Figure 5.10(b) and (c) is a typical response in initial sampling, showing the challenge in extracting FoMs as the performance is far from a practical inverter.

Therefore, in this study, raw inverter characteristics V are transformed into N extracted metrics compared with the ideal response V_{ideal}:

$$y_i = \text{MSE}(V^i - V^i_{\text{ideal}}) \quad i = 1, 2, \ldots, N \tag{5.4}$$

where MSE(·) is the mean square error function. For DC characteristics, the curve is divided into three parts to extract features, as enumerated and illustrated in Figure 5.10(a): (1) the high-level region, (2) the low-level region, and (3) the central switching point. Equation (5.4) is then applied to compute the metric value for each part. Similarly, for pulse characteristics, we derive four parts using the same idea as shown in Figure 5.10(b): (4) the high-level region, (5) the low-level region, (6) the rising edge, and (7) the falling edge, comparing with the steepest ideal switching (green dashed line) as well. For an ideal—or rather, theoretical—inverter, all features extracted would be zero initially. By using this approach, even when the circuit's operation state deviates significantly from a practical inverter, the above metrics can be used to discriminate candidate designs with different qualities.

5.4.2.3 Actor-critic-based Optimization

The actor-critic-based optimization is a quasi-reinforcement learning algorithm [237] developed from the DDPG algorithm [238], which is commonly used for continuous space control problems. Traditional reinforcement learning algorithms like DDPG are designed to train agents to operate in specific environments towards defined targets. However, these algorithms typically require thousands or even tens of thousands of operating trajectories for training, which is impractical for semiconductor devices and circuit design due to high computational costs. Additionally, algorithmic optimization in most contexts is a non-Markovian process, which violates the fundamental assumptions underlying reinforcement learning. Therefore, applying these reinforcement learning algorithms directly to such problems is inappropriate without adaptation. More discussion to clarify the connection and difference between reinforcement learning and optimization refers to Appendix. D. In general, actor-critic-based optimization is proposed for expensive optimization tasks, adapting the concepts of *actor* and *critic* from their traditional use in DDPG but with a different purpose. The algorithm trains a critic network based on the current dataset, uses actor network to exploit the promising design region over the model, and outputs the best-predicted design for the next evaluation. The main terms of the algorithm are first clarified and then the procedure is described in detail.

- **Design space:** This is the space of the design variables $\boldsymbol{x}$ for the inverter, which is continuous and bounded within given intervals.

- **Action space:** This refers to the space of the perturbations, or actions, $\boldsymbol{a}$ applied to the design variables, where $\boldsymbol{x} + \boldsymbol{a}$ forms a new design within the design space. The action space is continuous and typically bounded by twice the design intervals.
- **Data augmentation:** As the only available data for the training model are the design-metrics pairs extracted from the simulation during optimization, the data volume is quite limited. To mitigate training challenges, the design-metrics pairs are augmented into design-action-metrics triples. Given a set of design-metrics pairs $\mathcal{D} = \{(\boldsymbol{x}, \boldsymbol{y})\}$, the augmented dataset is $\mathcal{B} = \{(\boldsymbol{x}, \boldsymbol{a}, \boldsymbol{y}) | (\boldsymbol{x} + \boldsymbol{a}, \boldsymbol{y}) \in \mathcal{D}\}$. Obviously, if $\mathcal{D}$ contains N elements, then $\mathcal{B}$ contains N^2 elements.
- **Critic network:** A neural network $Q_\phi(\boldsymbol{x}, \boldsymbol{a})$, parameterized by ϕ, accepts a design $\boldsymbol{x}$ and an action $\boldsymbol{a}$, and predicts the performance of the design $\boldsymbol{x} + \boldsymbol{a}$. Q_ϕ is trained using the mean square error on a set of design-action-metrics triples by

$$L_\phi(\boldsymbol{x}, \boldsymbol{a}) = \left(Q_\phi(\boldsymbol{x}, \boldsymbol{a}) - y(\boldsymbol{x} + \boldsymbol{a})\right)^2 \tag{5.5}$$

  The critic network inherits its parameter ϕ from the last training, denoted by ϕ^-. The inheritance makes training faster and easier in a sequential manner.
- **Actor network:** A neural network $\mu_\theta(\boldsymbol{x})$, parameterized by θ, produces the most promising action for a given design $\boldsymbol{x}$. μ_θ is trained after the training of the critic network. The loss function aims to minimize the weighted sum of metrics estimated by the critic network, effectively exploring the design space while heading to promising regions of lower metrics. In addition, to ensure that the produced design $\boldsymbol{x} + \mu_\theta(\boldsymbol{x})$ stays within the given bounds, a penalty term is added to the loss function

$$L_\theta(\boldsymbol{x}) = w(Q_\phi(\boldsymbol{x}, \mu_\theta(\boldsymbol{x}))) + \psi(\boldsymbol{x} + \mu_\theta(\boldsymbol{x})) \tag{5.6}$$

  where $w(\cdot)$ is the weighted sum function and $\psi(\cdot)$ enforces the bound constraints defined by

$$\psi(\boldsymbol{x}) = \left\| \frac{\max\left(0, \boldsymbol{x}_{\text{low}} - \boldsymbol{x}\right)}{\boldsymbol{x}_{\text{high}} - \boldsymbol{x}_{\text{low}}} \right\|_2^2 + \left\| \frac{\max\left(0, \boldsymbol{x} - \boldsymbol{x}_{\text{high}}\right)}{\boldsymbol{x}_{\text{high}} - \boldsymbol{x}_{\text{low}}} \right\|_2^2$$

  where x_{low} and x_{high} are the lowest and highest values of the design in the dataset $\mathcal{D}$. $\max(0, \cdot)$ is the element-wise function that outputs a lager value compared to 0.

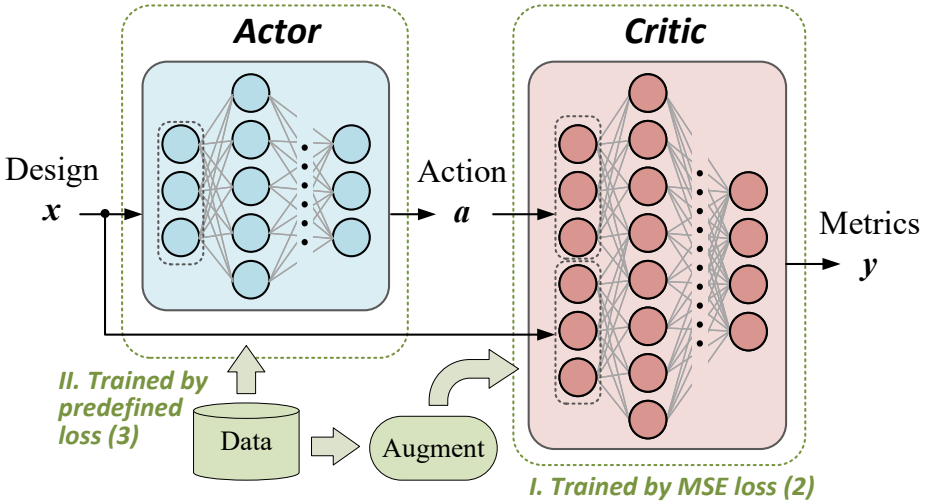


Figure 5.8: The structure of the actor and critic network as well as their training process.

Figure 5.8 outlines the structure of the actor and the critic network. Specifically, the actor network is employed to generate the best action over a given design and the critic network is employed as an estimator to evaluate how good the new design is. Both networks are sequentially stacked by a fully connected layer, batch-normalization layer, and rectified activation layer. The training algorithm is stochastic gradient descent.

Algorithm 9 Actor-critic-based optimization

Input: Actor network μ_θ, Critic network Q_ϕ, dataset $\mathcal{D}$, and inherited parameter ϕ^-(optional).

Initialize θ and ϕ
if ϕ^- is defined **then**
    $\phi \leftarrow \phi^-$
end if
Augment dataset $\mathcal{D}$ into $\mathcal{B}$
Train the critic network Q_ϕ on $\mathcal{B}$ by Equation (5.5)
Train the actor network μ_θ on $\mathcal{D}$ using Equation (5.6)
$\mathcal{Y} \leftarrow \{\}$
for each $(\boldsymbol{x}, \cdot) \in \mathcal{D}$ **do**
    $\boldsymbol{a} \leftarrow \mu_\theta(\boldsymbol{x}) + \epsilon_a(\boldsymbol{x}_{r_1} - \boldsymbol{x}_{r_2})$ ▷ Add noise on action.
    $\hat{\boldsymbol{y}} \leftarrow Q_\phi(\boldsymbol{x}, \boldsymbol{a})$
    if $\hat{\boldsymbol{y}} < \min(\mathcal{Y})$ **then**
        $\boldsymbol{x}^* \leftarrow \boldsymbol{x} + \boldsymbol{a}$
    end if
    $\mathcal{Y} \leftarrow \hat{\boldsymbol{y}} \cup \mathcal{Y}$
end for
$\phi^- \leftarrow \phi$

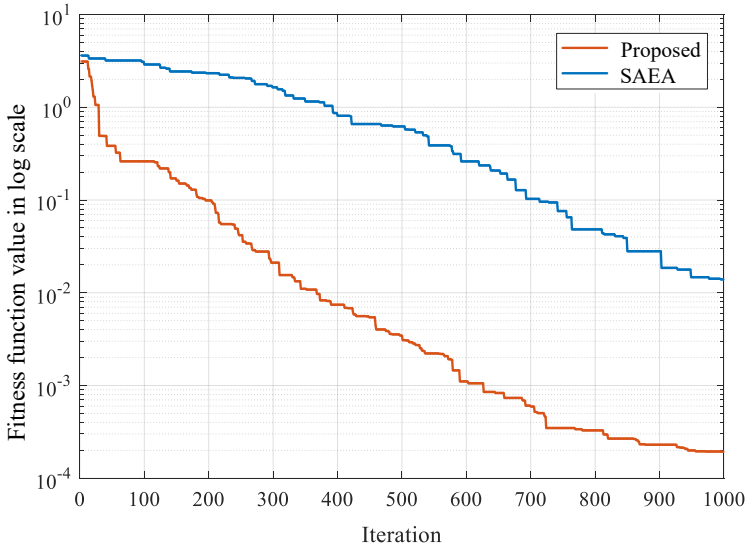
Output: Suggest the new design $\boldsymbol{x}^*$

The pseudo-code of the actor-critic-based optimization is shown in Algorithm 9. Following network training, each design in the current dataset is passed to the actor network to generate an action. The action is then added with the noise consisting of the difference between two randomly selected designs from the dataset, scaled by a hyper-parameter ϵ_a. The purpose is to balance the exploration and exploitation and enhance the algorithm's robustness. The new design added with the action is input into the critic network for querying metrics, and only the one with the best predictive metrics is output for simulation in the next iteration.

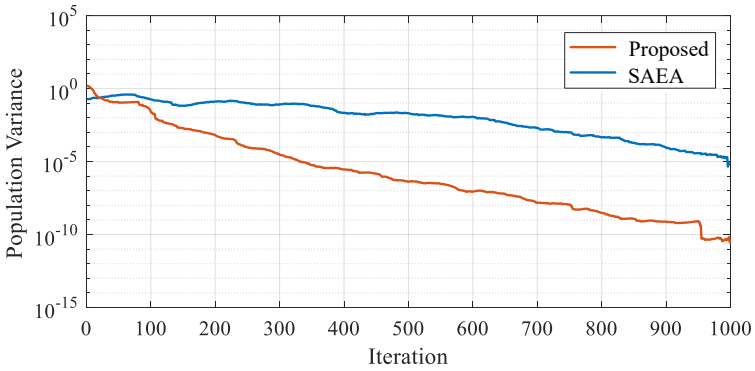
The number of hidden layers and neurons in each layer is set to ensure that the networks have sufficient modeling capability; therefore, these are generally determined by the number of parameters and metrics. In our experiment, the critic network has two hidden layers, each containing 16 neurons, while the actor network has three hidden layers, each containing 16, 23, and 16 neurons, respectively. In addition, ϵ_a is set to 0.1 for good balancing ability, while the number of initial samples is set to 25.

To verify the performance of the proposed optimization algorithm, several well-known mathematical benchmark functions with diverse landscapes are used. These include the Zakharov, Sphere, Rastrigin [239], Griewank, and Ackley [240] functions. All of these functions share a global minimum value of 0 located at $(0, \ldots, 0)$. For more information on these benchmark functions, please refer to Appendix A. The function values are normalized to a $[0, 1]$ range and then summed together to form the fitness, as formulated in Equation (4.1). A 20-dimensional problem was tested, with the search range defined as $[-20, 20]^{20}$. Both critic and actor networks consisted of two fully-connected hidden layers, each with 128 neurons. For comparison, an optimization algorithm used in Section 5.3 (Case Study 1) is employed. The function evaluation budget is set to 1000 iterations, with no initial solution provided.

The convergence results are shown in Figure 5.9(a), alongside the dataset (training) variance for each iteration. This figure shows that the proposed algorithm performs well on this benchmark problem, converging to 10^{-3} within 600 iterations. The results demonstrate that the proposed algorithm significantly improves efficiency, showcasing its strong ability to exploit the solution space, which is highly valuable in semiconductor design processes. Furthermore, Figure 5.9(b) shows that the algorithm maintains a lower variance in the dataset than the reference method, suggesting that it can efficiently handle complicated problems even with a smaller dataset diversity, which is another key advantage for device optimization.



(a) Convergence trend



(b) Population/Database variance

Figure 5.9: The convergence trend and population variance versus iterations on the benchmark problem

5.4.3 Result and Discussion

The evaluation of the CMOS inverter's performance was conducted through an analysis of its voltage transfer response and pulse transient response. To validate our methodology, two design cases in nanosecond and picosecond pulses are considered, respectively. The input pulse for the nanosecond case is 100 ns, while that for the picosecond case is 140 ps. These two-pulse cases are chosen based on [241]. Due to the limitation of the device's maximum oscillation frequency, meeting functional requirements with a nanosecond pulse for a CMOS inverter is relatively straightforward, while maintaining the same performance with a picosecond pulse is more challenging. In this case, the output signal may deviate significantly from the requirement, leading to additional delay, overshoot, and other performance degradation. Therefore, it was chosen as an excellent case to demonstrate how our methodology can help to find the optimal solution.

Considering manufacturability, the same gate lengths of NMOS and PMOS are kept for both cases, and the most flexible case in which gate lengths can be different is also provided to fully explore the design space. The resulting designs are compared with respect to transfer characteristics, providing much inspiration for the future technology node.

Table 5.7 lists the best parameter values, which were found within 100 iterations for the first case, and 200 iterations for the second case. More iterations are assigned to the second case to ensure convergence. All the parameter values are obtained by the algorithm to get the desired output characteristics close to the ideal one (the target data). Compared to other modeling-based methodologies [218, 232], our approach significantly reduces computational costs by eliminating the need for thousands of simulations required to train device models.

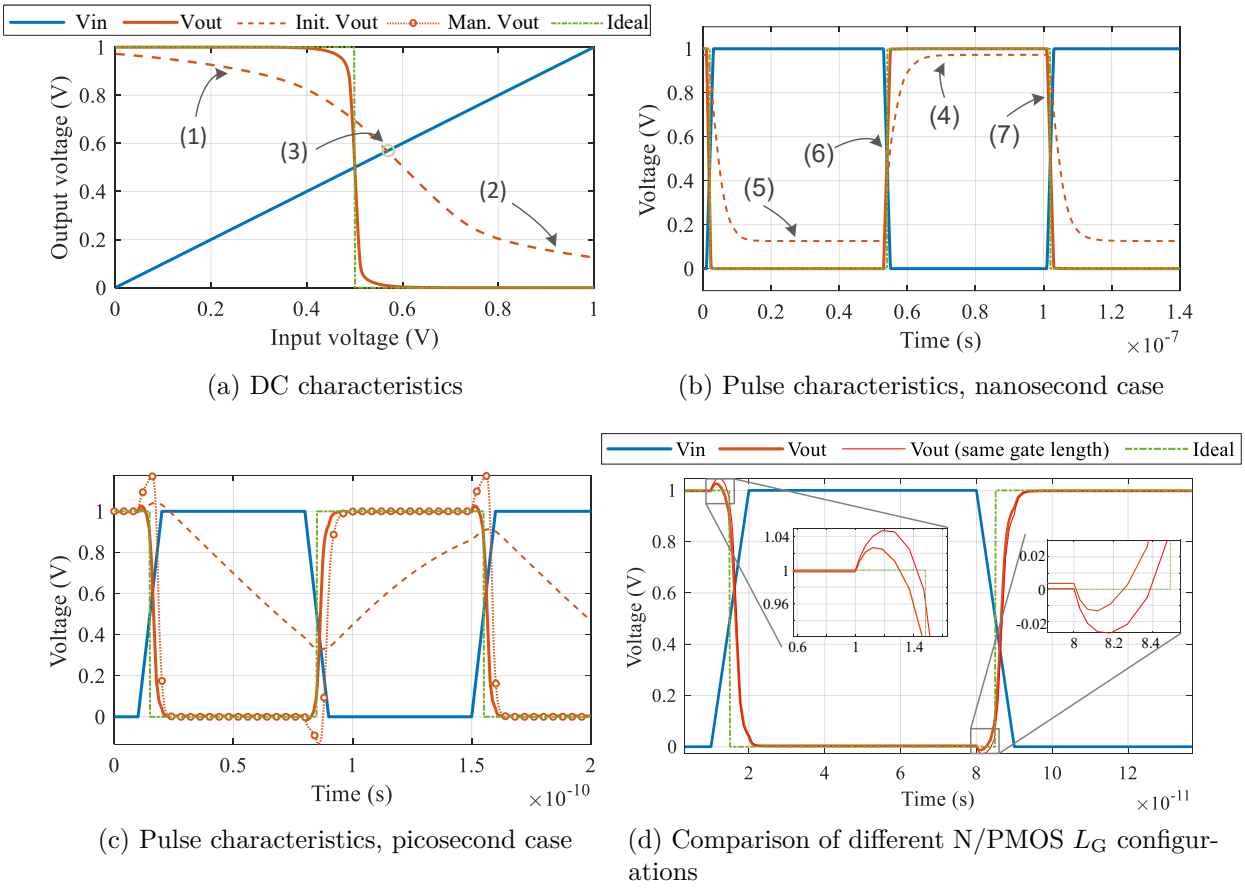
Table 5.7: Optimized parameter values for various cases

	Case 1 (ns pulse)		**Case 2 (ps pulse)**	
	NMOS	PMOS	NMOS	PMOS
L_G (nm)	58		24	
T_{Si} (nm)	24	27	12	13
N_{Ch} ($\times 10^{17}$cm^{-3})	3.8	5	0.5	0.7
W_{Si}	-	1.7	-	4
C_L (fF)	0.5		0.01	

	Case 2 (diff. L_G) (ps pulse)		
	NMOS	PMOS	Man. N/P
L_G (nm)	17	20	50
T_{Si} (nm)	12	15	20
N_{Ch} ($\times 10^{17}$cm^{-3})	2.6	2.3	5 / 6
W_{Si}	-	3.1	- / 2
C_L (fF)	0.01		0.01

The corresponding DC and dynamic characteristics are illustrated in Figure 5.10 by solid lines. As shown in Figure 5.10(a), the voltage transfer characteristic of the inverter reveals the expected inverting behavior. The input voltage V_{in} is plotted against the output voltage V_{out}, along with the typical initial and manual V_{out} points to depict the switching threshold. Three distinct regions can be observed as Region (1): this represents the cut-off region where V_{out} is high and almost equal to V_{DD}, indicating that the NMOS is in the OFF state and the PMOS is in the ON state; Region (2):

the saturation region where V_{out} approaches ground level, indicating that NMOS is on and PMOS is off; and Region (3): the transition region is characterized by a sharp fall in V_{out} as V_{in} increases. This region is crucial as it defines the inverter's switching threshold and gain.



(a) DC characteristics

(b) Pulse characteristics, nanosecond case

(c) Pulse characteristics, picosecond case

(d) Comparison of different N/PMOS L_G configurations

Figure 5.10: DC, pulse characteristics of two design cases, and comparison between the same and different N/PMOS L_G.

Voltage transfer characteristics indicate a superior switching behavior of the optimized inverter over the manually designed inverter with steeper slopes for both rising and falling edges. Region (1) and (2) indicate that the typical initial design's pull-up and pull-down functions are not as effective as those of the co-optimized design. The intersection point of the input and output curves, which ideally should be half V_{DD} for the inverter, is significantly different in the typical scenario as compared to the co-optimized design. This suggests a better noise margin and a more robust operation in the presence of voltage variations.

The pulse characteristics are compared with existing literature [241, 242] and a manually designed circuit. Table 5.8 provides a comprehensive comparison of figures of merit for CMOS inverters based on different technology nodes and design optimizations, including low-doped drain (LDD) FinFET and silicon on insulator (SOI) complementary FinFET (C-FinFET). The metric comparison includes rise time, fall time, edge rise, delay times, propagation delay, contamination delay, maximum oscillation frequency (MUF), and overshoot voltage.

The optimized inverter design using Planar FET technology shows superior performance compared to the manually optimized (Manual) Planar FET design across several parameters as shown in Figure 5.10(b) and (c). Specifically, the ML-assisted design demonstrates faster rise and fall times (4.20 ps and 3.6 ps respectively) compared to the manual design (7.95 ps and 8.22 ps respectively). This indicates a more rapid transition between logic states which is critical for high-speed applications.

Table 5.8: Comparison figures of merit ofdifferent inverters operating on picosecond pulse

	Co-opt.	**Manual**	**[242]**	**[241]**
Technology	Planar FET	Planar FET	SOI C-FinFET	LDD-FinFET
Supply voltage (V)	1.0	1.0	1.0	1.5
Signal Period (ps)	140	140	200	140
Rise time (ps)	4.20	7.95	10.00	4.56
Fall time (ps)	3.6	8.22	10.00	4.10
Edge rise (ps)	3.9	8.08	10.00	4.33
High to low delay (ps)	1.45	3.66	2.11	3.27
Low to high delay (ps)	1.52	4.5	1.49	3.55
Propagation delay (ps)	1.58	4.08	1.30	3.41
Contamination delay (ps)	1.45	3.66	0.9	3.27
MUF (THz)	0.12	0.06	0.05	0.11
Overshoot (V)	0.026	0.170	-	-

Voltage overshoot during the rise and fall period of the pulse is a critical parameter for signal integrity and reliability. This depends on the charging and discharging time of the output capacitor further determined by its time constant. When comparing MOS devices with the same and different gate lengths, as seen in Figure 5.10(d), we found that the voltage deviation at rise and fall periods is smaller when the gate length is used as an optimized parameter separately for N- and P-MOS devices than when the variable is constrained to the same value throughout the optimization. This is

because the gate length influences the fringing capacitance associated with the channel components as affects the intrinsic capacitance value of the device [241]. The ML-assisted design demonstrates a negligible undershoot and a minimal overshoot of 0.026 V which is significantly lower than the manual design's overshoot of 0.170 V. This suggests that the ML-assisted design is not only faster but also more precise, with less risk of damaging other components or causing logic errors due to excessive voltage. Hence, the proposed methodology helps find the most optimal parameter set and can be easily extended to complicated circuits and speed up the time-to-market of new devices.

To demonstrate the robustness of the optimization algorithm, more experiments with different configurations of algorithm parameters were conducted. The results indicate that the bound constraint $\psi(\cdot)$ in Equation (5.6) is indispensable. Without this constraint in the loss function of actor network, the algorithm suggests new designs that are impractical and far outside the given range. The settings for the number of hidden layers and neurons in each layer prove to be robust, provided the modeling capacity is sufficient. Therefore, configuring two or more hidden layers, with each layer having more than $2d$ neurons, where d is the number of neurons of the input layer, is adequate and has a limited impact on the final outcomes. The noise figure ϵ_a on the action balances efficiency and exploration; a larger value results in a slower convergence rate. For this optimization task, 0.1 was deemed appropriate through our experiments.

5.5 Summary

This chapter focuses on the algorithmic design methodology for semiconductor devices. It begins with a brief introduction to the devices and TCAD simulation. The concept of DTCO is introduced, along with a discussion of its limitation. By explaining implementation of the TCAD interface, two case studies based on TCAD simulation are presented. In the first case study, the structure of the epitaxial layer of an InP pHEMT is designed and optimized for terahertz operating frequency. Compared to the commercial pHEMT, the optimized design achieves 57% and 37% improvements in its cut-off frequency and maximum oscillation frequency, respectively, without altering the gate length. In the second case study, the concept of device circuit co-optimization is proposed and validated on a CMOS inverter. Using a novel actor-critic-based optimization algorithm, the proposed design methodology achieves better performance on

an inverter with planar FETs than that with advanced technology nodes, e.g., LDD FinFET and SOI C-FinFET. In summary, this chapter explores the potential pathway for algorithmic design methodology using TCAD simulation and highlights the use of machine learning in enhancing design automation within TCAD.

Chapter 6

Conclusions and Future Work

This chapter summarizes the conclusions drawn from Chapter 3 to Chapter 5 of this thesis. The chapter also discusses and highlights potential routes for further exploration and expansion from the current research.

In general, this thesis investigates the potential path toward machine learning-assisted EDA for distributed-element devices/circuits and semiconductor devices. Due to computationally expensive simulation, design automation in these areas is considered challenging and intractable. The circuits and devices selected for research in this thesis are both relevant and diverse. From the perspective of electronic engineering, the selected circuits include passive, active, and transistor components. The simulation complexity and time cost also increase accordingly. From the perspective of optimization algorithms, the design problems of microwave filters are very challenging due to their multimodal landscapes. The challenge of designing power amplifiers mainly lies in their problem scales, i.e., the multivariate performance metrics and design variables. And the design optimization of semiconductor devices demands high efficiency. Therefore, this research can be viewed as a comprehensive EDA research to address problems within the areas mentioned above. With these distinctions, each area is discussed in detail, respectively.

6.1 Microwave Filter Design Automation

For the research on microwave filter design automation presented in Chapter 3, an unsupervised design methodology is proposed, consisting of a systematic sampling method with two-phase design optimization. Once the initial design is constructed via a programmable approach, the entire process can produce the final physical implementation without human intervention. The novel objective functions employed in different optimization phases successfully capture landscape structures, leading to better solutions. Payoffs of the proposed methodology are validated and compared through two real-world examples. Furthermore, it is concluded that design knowledge, such as positions of zeros and poles and resonator theory, plays an important role in achieving better outcomes. Without considering design knowledge, such as the resonator theory in the sampling method, both effectiveness and success rate degrade. From the perspective of algorithms, global optimization with the assistance of GP surrogates is advantageous in finding optimal solution but struggles with limited efficiency. By incorporating an appropriate local search strategy, the entire framework demonstrates potent capability in searching complex landscapes efficiently.

In addition, the proposed methodology is not limited to any specific filter types. Two examples with transmission zeros and a dozen variables have been demonstrated, showing competitive performance compared to other methods. When more complex design cases, such as higher-order filters with tens of design variables, are considered, they can be addressed by tuning some hyperparameters of the algorithm to allow for greater exploration capacity. Therefore, this methodology demonstrates broader applicability in the field of filter design automation.

An extension of this research is to investigate yield optimization considering manufacturing tolerance. This is a crucial follow-up step after design optimization, which accounts for process errors in design to maximize yield, and should be developed independently. Another extension is to promote the investigation into the design of diplexers. Diplexers can be viewed as devices composed of multiple microwave filters. The current methodology requires designing each branch separately and then combining them for manual tuning; therefore, achieving unsupervised diplexer design is highly meaningful, although very challenging. Furthermore, research on algorithms and machine learning techniques is also worth exploring. Investigating the use of generative AI for filter topology design could significantly advance filter design automation in the future.

6.2 MMIC Power Amplifier Design Automation

For the research on power amplifiers presented in Chapter 4, a new methodology targeting general layout-level PA design automation is proposed, which consists of an optimization-oriented integrated environment and a BNN-based optimization algorithm. The proposed methodology enables a higher degree of MMIC PA design automation, validated by two practical and challenging examples. The optimization-oriented integrated environment bridges external algorithms with ADS, making the methodology able to manage commercial PDKs and compatible with current design workflows. By incorporating Bayesian neural networks, the proposed methodology can predict and identify the most promising solution in each iteration and achieve reasonable efficiency with around 500 simulation runs on the presented examples. In addition, it is found that the proposed methodology is potent for the cases requiring performance consistency, which is often difficult to achieve through manual trade-offs across the entire band.

Similar to the design automation of microwave filters, yield optimization is also crucial for MMIC PAs. However, considering manufacturing tolerance in MMICs is more challenging and even intractable due to multiple sources of variability, such as the accuracy of PDKs and process variations in transistors and passive components. Therefore, it should be explored as a separate research topic, known as design for manufacturing.

In terms of potential future research, several directions are suggested. First, although this research has promoted considerable progress toward PA design automation, an unavoidable issue is that the current methodology lacks the ability to predict the possible performance limit. In other words, it is impossible to know whether the given specifications can be achieved or not before the optimization is completed. When the given specifications are difficult to achieve, the output best design may not meet expectations or even be misled. A feasible approach to mitigate this issue is to run the proposed algorithm on schematics first. The optimal performance achievable by schematics can generally be considered the performance upper limit involving layout (with EM simulation). More solutions are worth exploring. Second, the methodology can be extended to multi-objective optimization. Typically, there is a conflict between PA efficiency and output power or gain and linearity. When a set of trade-off designs is needed rather

than a single optimal design, a methodology based on multiobjective optimization is more appropriate. Third, an expansion of this research is to investigate topology design automation. As discussed in the main body, this is also an interesting and attractive problem to solve.

6.3 Algorithmic Design Optimization for Semiconductor Devices

The research in Chapter 5 promotes an attempt toward algorithmic design optimization for semiconductor devices. By implementing a TCAD interface, algorithms are able to interact with TCAD simulation for device optimization. Two case studies are conducted to manifest the effect of this approach. The first case study uses GP-based SAEA to optimize the epitaxial layer structure of a pHEMT for higher cut-off frequency and maximum oscillation frequency. Significant improvements were achieved, with a 57% improvement in the cut-off frequency and a 30% increase in the maximum oscillation frequency compared to the commercial design. The optimization process took just 16 hours on a standard desktop computer, achieving a substantial reduction in time consumption compared to the trial-and-error method. For the second case study, the concept of device circuit co-optimization is proposed, incorporating a novel actor-critic-based optimization algorithm. The effectiveness of the proposed methodology is validated by a CMOS-based inverter, achieving better performance regarding picosecond switching characteristics on a planar N/PMOS than with advanced technology. In summary, this research demonstrates that using machine learning techniques to enable algorithmic design optimization within TCAD can bring considerable benefits, which pave the way for future research on more complex circuit and design challenges.

A straightforward extension from this research is to apply the proposed methodology to more advanced technology and complex circuits, such as FinFET, Gate all around (GAA) FET, and nanosheet architecture. The structure of these devices is more complex than planar CMOS, resulting in higher computational costs. This promotes new challenges for the efficiency of optimization algorithms and is indeed a promising direction as the next plan. In addition, process variability plays a significant role in the characterization of semiconductor devices. Therefore, incorporating variability-aware

models into the optimization framework to account for process-induced variations is a crucial step toward a robust and practical methodology. This prospective research will make algorithmic design optimization more applicable to real-world manufacturing scenarios.

Appendices

A Benchmark Functions

Here the benchmark functions used in this thesis are formulated and described in detail. In each function, d refers to the dimension of the argument and is often set to 20 in the numerical tests of the main body of this thesis.

- **Zakharov function**: a plate-shaped surface function that has different gradient values in different dimensions.

$$f(\boldsymbol{x}) = \sum_{i=1}^{d} x_i^2 + \left(\sum_{i=1}^{d} 0.5ix_i\right)^2 + \left(\sum_{i=1}^{d} 0.5ix_i\right)^4 \tag{1}$$

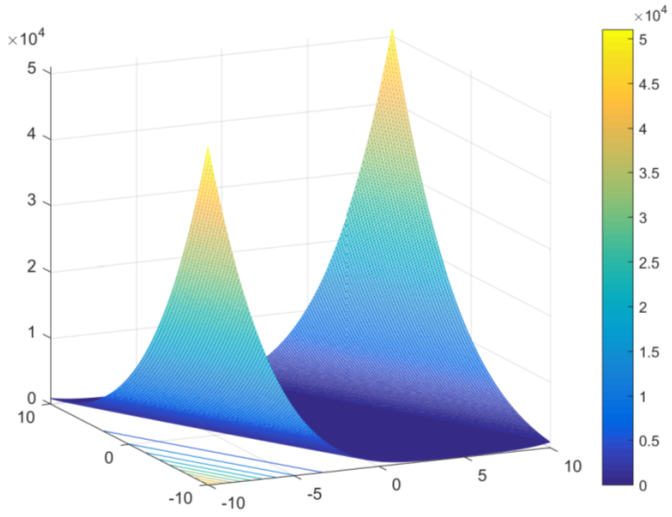


Figure A.1: Zakharov function in two dimensions

- **Sphere function**: a commonly used bowl-shaped function that is continuous, convex, and unimodal.

$$f(\boldsymbol{x}) = \sum_{i=1}^{d} x_i^2 \tag{2}$$

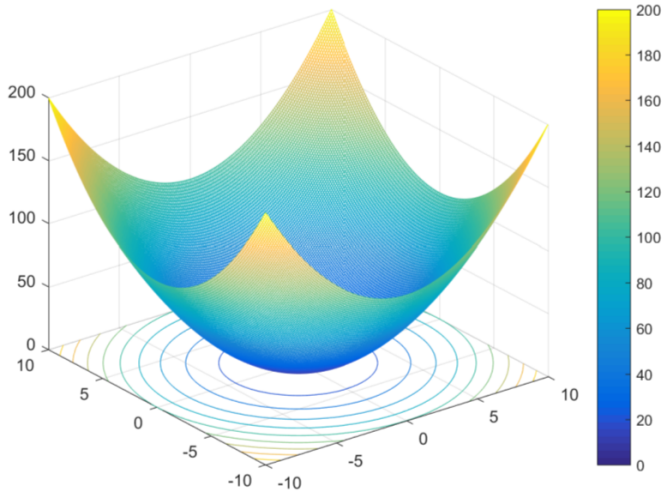


Figure A.2: Sphere function in two dimensions

- **Rastrigin functio**n: a function has several local minima and is highly multimodal, but the locations of the minima are regularly distributed.

$$f(\boldsymbol{x}) = 10d + \sum_{i=1}^{d} \left[x_i^2 - 10\cos\left(2\pi x_i\right)\right] \tag{3}$$

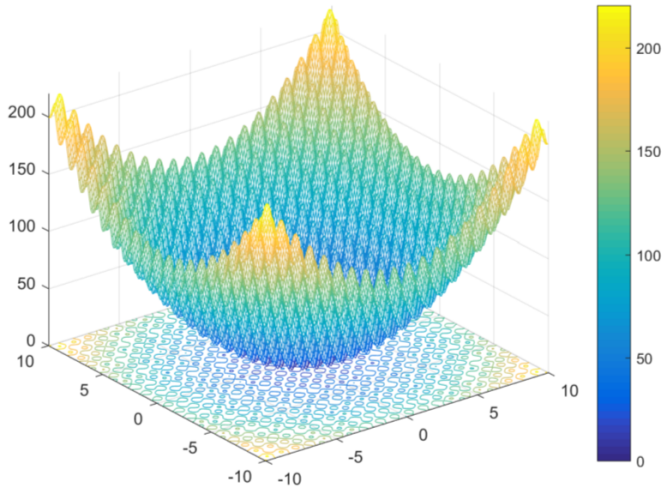


Figure A.3: Rastrigin function in two dimensions

- **Griewank function**: a highly rugged multimodal function that has many widespread local minima, which are regularly distributed.

$$f(\boldsymbol{x}) = \sum_{i=1}^{d} \frac{x_i^2}{4000} - \prod_{i=1}^{d} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \tag{4}$$

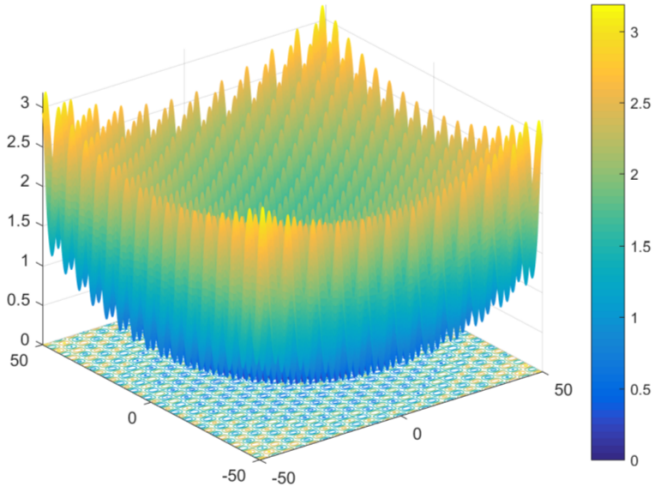


Figure A.4: Griewank function in two dimensions

- **Ackley Function**: a widely used multimodal function for testing. In its two-dimensional form, it is characterized by a nearly flat outer region and a large hole at the center. The function poses a great risk for optimization algorithms to be trapped in one of its many local minima.

$$f(\boldsymbol{x}) = -a \exp\left(-b\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos\left(cx_i\right)\right) + a + \exp(1) \qquad (5)$$

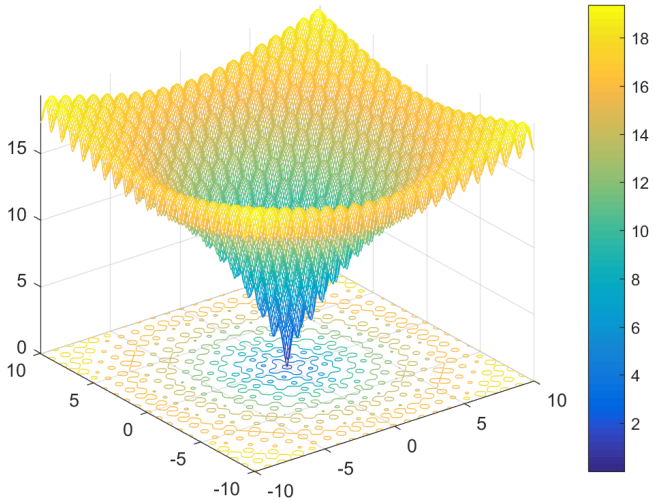


Figure A.5: Ackley function in two dimensions, when $a = 20, b = 0.2$ and $c = 2\pi$

B File and Code Example of Integrated Simulation Environment with ADS

Here gives the content of a typical high-level AEL script, used in simulation-oriented intergated environment.

```
if (ael_file_exists(fix_path("./lyxAutoRunADS.atf")))
{
    decl glob_sim_runing_status = TRUE;
    load(fix_path(".\\lyxAutoRunADS.atf"));
    lyx_open_design(fix_path("C:\Users\xxxxxxxxx\Documents\Project\
   PA_Design\210326_6x80_D01GH_AB_2W_wrk"),
                            "6x80_D01GH_AB_2W_lib:
   Complete_EM_test_OPT_N1:schematic");
    lyx_update_parameters(list("VAR2","VAR3","VAR6"),
                          list(list("x1","x2","x3","x4","x5","x6","x7","
   x8"),list("x9","x10","x11","x12","x13","x14","x15","x16"),list("x17","
   x18","x19","x20","x21","x22","x23","x24")),
                          list(list
   (3.5,72.3,277,159.6,3,1276.9,6.5,265.7),list
   (22,247.5,43,225.3,3,215.9,34,209.8),list
   (49.5,137,121.3,100.5,119.3,264.5,88,187.5)));
    remove("./objective.txt");
    lyx_run_sim_and_exist("6x80_D01GH_AB_2W_lib:Complete_EM_test_OPT_N1:
   schematic");
}
```

A code snippet of the simulation-oriented intergated environment implemented by MATLAB is presented.

```
function value = RunADSSimulation(ProjectPath, LibName, CellName,
   Parameters, time_estim)
file_id = DirectorySetup();
[work_path,~,~] =  fileparts(mfilename('fullpath'));
[project_root,project_name,~] = fileparts(ProjectPath);
% lib_name = [project_name(1:end-3),'lib'];
full_cell_name = sprintf('%s:%s:%s',LibName,CellName,'schematic');
%% prepare ael script
ael_content = fileread(fullfile(work_path,'testTemplate.ael'));
ael_content = strrep(ael_content,'_PROJECT_PATH_',ProjectPath);
ael_content = strrep(ael_content,'_CELL_FULL_NAME_',full_cell_name);
[comp_str, name_str, value_str] = ParseParameters(Parameters);
ael_content = strrep(ael_content,'_COMP_ITEM_',comp_str);
ael_content = strrep(ael_content,'_NAME_LIST_',name_str);
```

```
ael_content = strrep(ael_content,'_VALUE_LIST_',value_str);
ael_path = fullfile(work_path,[file_id,'.ael']);
log_path = fullfile(ProjectPath,'data',[CellName,'_data'],'logFile.txt');
SaveAEL(ael_path,ael_content);
%% run simulation
ads_path = 'ads';
ads_command = sprintf('%s -m %s',ads_path,ael_path);
check_file_path = fullfile(ProjectPath,'data',[CellName,'.ds']);
tic
err_count = 0;
while err_count < 10
    try
        cleanUpResults(check_file_path);
        runADS(ads_command, check_file_path,time_estim);
        log_str = fileread(log_path);
        if contains(log_str,'Error')
            SaveAEL(fullfile(fullfile(work_path,'.log',[file_id,'_log.txt
    '])),log_str);
            pause(2);
            error('Error log saved to .log folder.');
        end
    break;
    catch
        err_count = err_count + 1;
    disp('Error occurred, retry it now...');
    end
end
toc
pause(3);
%% fatch results
textdata_path = fullfile(work_path,'.dsdata',[file_id,'_text.txt']);
dsdump_command = sprintf('%s %s > %s','dsdump',check_file_path,
    textdata_path);
runADS(dsdump_command,textdata_path,10);
obj_path = fullfile(work_path,'.obj',[file_id,'_obj.txt']);
copyfile(fullfile(ProjectPath,'objective.txt'),obj_path);
value = csvread(obj_path);
movefile(ael_path,fullfile(work_path,'.ael'));
end

function cleanUpResults(check_file_path)
try
    delete(check_file_path);
catch
    return;
end
end
```

```
59
60 function runStatus = runADS(ads_command, res_monitor, time_estim)
61 tic
62 system(ads_command,'-echo'); pause(2);
63 while toc < time_estim
64     if exist(res_monitor,'file')
65         runStatus = 0;
66        %system('taskkill /f /im "hpeesofde.exe"')
67         return;
68     end
69     pause(2);
70 end
71 if ~exist(res_monitor,'file')
72     error('Run command error, no file outputs.');
73 end
74 end
75
76 function SaveAEL(SavePath, Content)
77 fid = fopen(SavePath,'w');
78 fprintf(fid,'%s',Content);
79 fclose(fid);
80 end
81
82 function [comp_str, name_str, value_str] = ParseParameters(Parameters)
83 comp_str = jsonencode(Parameters.comp);
84 comp_str = comp_str(2:end-1);
85 name_str = jsonencode(Parameters.name);
86 name_str = name_str(2:end-1);
87 if length(Parameters.comp)==1
88     value_str = num2str(Parameters.value,'%.2g,');
89     value_str = value_str(1:end-1);
90 else
91     name_str = strrep(name_str,'[','list(');
92     name_str = strrep(name_str,']',')');
93     comp_str = ['list(',comp_str,')'];
94     Parameters.value = RoundCell(Parameters.value,2);
95     value_str = jsonencode(Parameters.value);
96     value_str = value_str(2:end-1);
97     value_str = strrep(value_str,'[','list(');
98     value_str = strrep(value_str,']',')');
99 end
100 end
101
102 function outcell=RoundCell(incell,n)
103 outcell = cell(size(incell));
104 for i=1:length(incell)
105     outcell{i}=round(incell{i},n);
```

```
106 end
107 end
108
109 function file_id = DirectorySetup()
110 persistent sub_idx;
111 if isempty(sub_idx)
112     sub_idx = 1;
113 end
114 file_id = [datestr(datetime,'yymmddHHMM-'),num2str(sub_idx)];
115 fprintf('Project id: %s ',file_id);
116 if ~exist(fullfile(pwd,'.ael'),'dir')
117     mkdir(fullfile(pwd,'.ael'));
118 end
119 if ~exist(fullfile(pwd,'.dsdata'),'dir')
120     mkdir(fullfile(pwd,'.dsdata'));
121 end
122 if ~exist(fullfile(pwd,'.fig'),'dir')
123     mkdir(fullfile(pwd,'.fig'));
124 end
125 if ~exist(fullfile(pwd,'.obj'),'dir')
126     mkdir(fullfile(pwd,'.obj'));
127 end
128 if ~exist(fullfile(pwd,'.log'),'dir')
129     mkdir(fullfile(pwd,'.log'));
130 end
131 if ~exist(fullfile(pwd,'testTemplate.ael'),'file')
132     CreatAELTemplate();
133 end
134 sub_idx = sub_idx+1;
135 end
136
137 function TemplateStr=CreatAELTemplate()
138 TemplateStr=['if (ael_file_exists(fix_path("./lyxAutoRunADS.atf")))\n'
    ...
139 '{\n' ...
140 '\tload(fix_path(".\\\\lyxAutoRunADS.atf"));\n' ...
141 '\tlyx_open_design(fix_path("_PROJECT_PATH_"),"_CELL_FULL_NAME_");\n' ...
142 '\tlyx_update_parameters(_COMP_ITEM_,list(_NAME_LIST_),list(_VALUE_LIST_)
    );\n' ...
143 '\tlyx_run_sim_and_exist("_CELL_FULL_NAME_");\n' ...
144 '}\n'];
145 fid = fopen(fullfile(pwd,'testTemplate.ael'),'w+');
146 fprintf(fid,TemplateStr);
147 fclose(fid);
148 end
```

C File and Code Example of TCAD Interface

Here shows the content of a typical `gtree.dat` file.

```
# Copyright (C) 1994-2011 Synopsys Inc.
# swbtree vcurrent, Tue Mar 30 07:24:56 2021

# --- simulation flow
sde1 sde "-h 10240" {}
sde1 Hbuffer1 "0.03" {0.3}
sde1 Hchannel1 "0.01" {0.01}
sde1 Hbarrier1 "0.01" {0.01}
sde1 Hspacer1 "0.003" {0.004}
sde1 Hcap1 "0.0085" {0.0085}
sde1 SheetChargeSpacer "7e12" {7e+12}
sde1 SheetChargeBarrier "4e12" {4e+12}
sde1 SheetChargeCap "1e13" {1e+13}
sdevice3 sdevice "" {}
sdevice3 Vg "-1.2" {0.2}
Fmax2 svisual "" {}
Fmax2 P "MAG" {MAG}
svisual4 svisual "" {}
RF2 svisual "" {}
# --- variables
# --- scenarios and parameter specs
scenario default Hbuffer1 ""
scenario default Hchannel1 ""
scenario default Hbarrier1 ""
scenario default Hspacer1 ""
scenario default Hcap1 ""
scenario default SheetChargeSpacer ""
scenario default SheetChargeBarrier ""
scenario default SheetChargeCap ""
scenario default Vg ""
scenario default P ""
# --- simulation tree
0 1 0 {} {default} 0
1 2 1 {0.3} {default} 0
2 3 2 {0.01} {default} 0
3 4 3 {0.01} {default} 0
4 5 4 {0.004} {default} 0
5 6 5 {0.0085} {default} 0
6 7 6 {7e+12} {default} 0
7 8 7 {4e+12} {default} 0
8 9 8 {1e+13} {default} 0
9 10 9 {} {default} 0
```

```
10 11 10 {0.2} {default} 0
11 12 11 {} {default} 0
12 13 12 {MAG} {default} 0
13 14 13 {} {default} 0
14 15 14 {} {default} 0
```

Here gives a code snippet of the TCAD interface.

```
function T = f_eval_obj_withpath(x,gtree_file,project_path,
    export_file_list)
uuid = datestr(datetime,'yymmddHHMMSS');
disp(['Project ID: ',uuid]);
%% Generate gtree.dat
% load x;
[work_path,~,~] =  fileparts(mfilename('fullpath'));
gtree_content = fileread(fullfile(work_path,gtree_file));
for i = 1:length(x)
    Vstr = ['_V',num2str(i),'_'];
    gtree_content = strrep(gtree_content,Vstr,num2str(x(i),'%.3g'));
end
fid = fopen(fullfile(work_path,'gtree.dat'),'w');
fprintf(fid,'%s',gtree_content);
fclose(fid);
disp('Build gtree.dat successfully.');
disp(gtree_content); pause(2);
%% run Simulation
% project_path = '/home/tcad2017/STDB/AI_RF_20210527/';
% export_file_list = {'n23_Gain.csv','n24_Pdcmin.csv','n25_NFmin.csv'};
runSimulationLocal(project_path,work_path,export_file_list);

%% read results
T = cell(1,length(export_file_list));
for i = 1:length(export_file_list)
    filepath = fullfile(work_path,export_file_list{i});
    if exist(filepath,'file')
        T{i} = readmatrix(filepath);
        movefile(fullfile(work_path,export_file_list{i}),...
            fullfile(work_path,'res',[export_file_list{i},'-',uuid]));
    else, T{i} = [];
    end
end

end

%% runSimulationLocal
function runSimulationLocal(remote_path,work_path,file_list)
```

```
clean_up_str = sprintf('gcleanup -d %s',remote_path);
runCommand(clean_up_str);
commond_str = sprintf('cp gtree.dat %s',remote_path);
runCommand(commond_str);
commond_str = sprintf('gsub -e all %s',remote_path);
try
    runCommand(commond_str);
catch
    clean_up_str = sprintf('gcleanup -d %s',remote_path);
    runCommand(clean_up_str);
    runCommand(commond_str);
end
% fatch results
for i=1:length(file_list)
    file_name = file_list{i};
    file_path = fullfile(remote_path,file_name);
    res_path = work_path;
    commond_str = sprintf('cp -f %s %s',file_path,res_path);
    runCommand(commond_str);
end
% disp('Run simulation successfully!');
end

%% runSimulationViaSSH
function runSimulationViaSSH(remote_path,local_path,file_list,user,ip)
% clean up
% disp('2. Clean up previous computation.');pause(2);
clean_up_str = sprintf('ssh %s@%s "gcleanup -d %s"',user,ip,remote_path);
runCommand(clean_up_str);
% tansfer file
commond_str = sprintf('scp -rp gtree.dat %s@%s:%s',user,ip,remote_path);
runCommand(commond_str);
% run simulation
run_solver_str = sprintf('"gsub -e all %s"',remote_path);
% commond_str = sprintf('ssh -o ServerAliveInterval=300 %s@%s %s',user,ip
    ,run_solver_str);
commond_str = sprintf('ssh %s@%s %s',user,ip,run_solver_str);
% disp('3. Run simulation.');pause(2);
try
    runCommand(commond_str);
catch
    clean_up_str = sprintf('ssh %s@%s "gcleanup -d %s"',user,ip,
    remote_path);
    runCommand(clean_up_str);
    runCommand(commond_str);
end
% fatch results
```

```
% disp('4. Fatch results.');pause(2);
if remote_path(end) ~= '/', remote_path=[remote_path,'/']; end
for i=1:length(file_list)
    file_name = file_list{i};
    file_path = [remote_path,file_name];
    commond_str = sprintf('scp %s@%s:%s %s',user,ip,file_path,local_path)
    ;
    runCommand(commond_str);
end
% disp('Run simulation successfully.');
end


function cmstat = runCommand(commond_str)
count_num = 0;
err_num = 0;
while count_num == err_num && err_num<10
    try
        [cmstat,cmlog] = system(commond_str);
%         fprintf('run commond successfully with status %d.',cmstat);
    catch
        err_num = err_num+1;
        disp('Error occurred, retry it now...');
    end
    count_num = count_num+1;
end
if err_num >= 10
    error('Run fail.')
end
end
```

D Discussion: Reinforcement Learning and Optimization

Recent developments in reinforcement learning (RL) have proven its capability in plenty of human-level works, including but not limited to the game of Go [243], complex video games [244], automatic drive [245], and control problems [246]. Derived from Markov decision processes (MDPs), although RL research has gone through a long history, its emphatic power has only emerged with the help of deep neural networks in recent decades. In 2014, Deepmind published their research on playing Atari [244] and declared the first deep reinforcement learning algorithm that outperforms all previous approaches and surpasses a human expert, opening up a new era of artificial intelligence. The DNN used in this algorithm was later called deep Q-learning networks (DQNs) and has also become a classical method in modern deep reinforcement learning. The subsequent research utilizing DNNs as a bunch of functional operators, like feature extractor, function approximator, probability estimator, or category classifier, provides plentiful remarkable variants in broad applications.

The prosperity of RL encourages more researchers to dive into this field. Some of them perceive the connection between RL and optimization algorithms (although RL itself uses stochastic gradient descent as the optimizer, it is not the purpose of RL) and start applying RL in solving complex combinatorial problems, i.e., NP-hard problems. [247, 248, 249, 250, 251, 252, 253] A couple of publications show that, compared to the off-the-shelf heuristics, RL-based algorithms are more robust and can be trained on small-scale training sets and applied to mediate new cases. A trained RL model can also cooperate with heuristics and obtain a better solution in less time than before. It is quite a promising area, however, such algorithms for combinatorial problems are not suitable for the problems discussed in this thesis, since almost all optimization problems are with the continuous domain.

The main characters of RL are the agent and the environment, as a comparison, the main characters of optimization are the optimizer and the problem. In RL, the environment is the world that the agent lives in and interacts with. At every step of the interaction, the agent takes action by its policy rule to the environment, observes the state transition, and receives the reward. The observation of the state may be partial, and the environment may be non-deterministic. The goal of the agent is to update its internal policy rule to tackle the environmental state transition and maximize the

cumulative reward, also called return. Since the cumulative reward involves future information, the agent must have the ability to forecast the expected reward from the future and determine whether the current action is good enough. That is the difficulty when considering the complicated action and state interactions.

As for the optimization, there is no concept of cumulative reward, while the objective function of the problem is solely concerned. At every iteration, the optimizer suggests new solutions and receives the corresponding function values (zero-order) from the problem. In some cases, gradient information (first-order) is also available, which will help determine the solution more effectively. Differing from RL, optimization algorithms, such as the Newton method, BFGS, and conjugate gradient method, always have rigorous theories on convergence. This ensures the feasibility of applying algorithms to various practical problems. Heuristic algorithms like evolutionary algorithm and particle swarm optimization, have become popular in recent decades for their ability to global exploration. A more general high-level term, metaheuristics, aims to seek, generate, or select a heuristic strategy that provides a sufficiently good solution to the problem. The core of metaheuristics is the search strategy, while the main difficulty lies in how to design the strategy for a balanced exploration and exploitation ability.

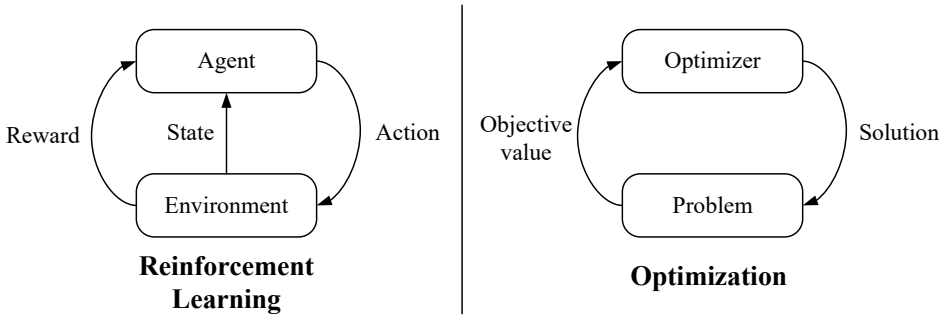


Figure D.1: Reinforcement Learning versus Optimization

As far as we know, there is no direct connection between RL and optimization. But if you view the optimizer as an agent and the problem as the environment, where the purpose of the optimizer is to explore the problem space and obtain a better solution, the two things look somewhat similar, as shown in Figure D.1. Thereafter, the solution can be seen as an action, and the objective value or any other information from the problem can be seen as the state. Although this view bridges the gap between the two concepts and is worthwhile to leverage from each other in both fields, it is far from mature to widely apply. Some obvious drawbacks and open questions are listed here:

- RL needs tens to hundreds of thousands of episodes for training, which means it is more computationally complex than heuristics for a given problem.

- Although, after training, the inference cost is relatively low, it is not clear to what extent the agent can be applied to new problems. In other words, the generalization ability of an agent is doubtful.
- It is not adequate to only regard the objective function values as states, as such a state may correspond to more than one action resulting in the action space being much larger than the state space. In most cases, there is only one optimal action. When optimization problems are black-box, this issue becomes more salient since there is no information available other than the objective value from problems. As we have discussed, design automation problems are such a type of black-box optimization problem.

In summary, this appendix briefly reviews the development of reinforcement learning in recent decades, and discusses the connection and differences between it with optimization, aiming to provide a more comprehensive understanding of why reinforcement learning algorithms should be adapted to specific optimization problems. Due to the intensive need for training data and different application scenarios, reinforcement learning algorithms are unlikely to be used directly for the problems discussed in this thesis.

Bibliography

[1] Laung-Terng Wang, Yao-Wen Chang and Kwang-Ting (Tim) Cheng. *Electronic Design Automation: Synthesis, Verification, and Test*. Morgan Kaufmann, Mar. 2009. ISBN: 978-0-08-092200-3.

[2] Louis Kossuth Scheffer, Luciano Lavagno and Grant Martin, eds. *Electronic Design Automation for IC System Design, Verification, and Testing*. Electronic Design Automation for Integrated Circuits Handbook. Boca Raton, FL: CRC Taylor & Francis, 2006. ISBN: 978-0-8493-7923-9.

[3] Luciano Lavagno et al. *Electronic Design Automation for IC Implementation, Circuit Design, and Process Technology: Circuit Design, and Process Technology, Second Edition*. second edition. Boca Raton: CRC Press, 2016. ISBN: 978-1-4822-5461-7.

[4] Karol Struniawski, Aleksandra Konopka and Ryszard Kozera. 'Exploring Apple Silicon's Potential from Simulation and Optimization Perspective'. In: *International Conference on Computational Science*. Springer. 2024, pp. 35–42.

[5] J. Darringer et al. 'EDA in IBM: Past, Present, and Future'. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 19.12 (Dec. 2000), pp. 1476–1497. ISSN: 02780070. DOI: 10.1109/43.898827.

[6] Zhengqi Gao and Duane S. Boning. *A Review of Bayesian Methods in Electronic Design Automation*. Mar. 2023. arXiv: 2304.09723 [stat].

[7] John W. Bandler and Jose Ernesto Rayas-Sanchez. 'An Early History of Optimization Technology for Automated Design of Microwave Circuits'. In: *IEEE Journal of Microwaves* 3.1 (Jan. 2023), pp. 319–337. ISSN: 2692-8388. DOI: 10.1109/jmw.2022.3225012.

[8] N. Loubet et al. 'Stacked nanosheet gate-all-around transistor to enable scaling beyond FinFET'. In: *2017 Symposium on VLSI Technology*. 2017, T230–t231. DOI: 10.23919/vlsit.2017.7998183.

[9] M.B. Steer, J.W. Bandler and C.M. Snowden. 'Computer-Aided Design of RF and Microwave Circuits and Systems'. In: *IEEE Transactions on Microwave Theory and Techniques* 50.3 (Mar. 2002), pp. 996–1005. ISSN: 00189480. DOI: 10.1109/22.989983.

[10] Tony Chan Carusone, David Johns and Kenneth William Martin. *Analog Integrated Circuit Design*. 2. ed. Hoboken, NJ: Wiley, 2012. ISBN: 978-0-470-77010-8.

[11] Synonsys. *Synonsys HSPICE*. https://www.synopsys.com/content/dam/synopsys/verification/datasheets/hspice-ds.pdf. [Online]. 2024.

[12] Cadence. *Cadence Spectre X Simulator*. https://www.cadence.com/zh_CN/home/resources/white-papers/revolution-by-evolution-getting-to-the-next-technology-breakthrough-in-analog-simulation-wp.html. [Online]. 2024.

[13] Anders Bondeson, Thomas Rylander and Pär Ingelström. *Computational electromagnetics*. Springer, 2012.

[14] Giuseppe Pelosi, Antonio Savini and Stefano Selleri. 'A Brief History of Computational Electromagnetics'. In: *2021 7th IEEE History of Electrotechnology Conference (HISTELCON)*. Ieee. 2021, pp. 81–84.

[15] Keysight. *Momentum Key Features*. https://www.keysight.com/zz/en/lib/resources/training-materials/momentum-key-features-1936424.html. [Online].

[16] C. K. Maiti. *Introducing Technology Computer-Aided Design (TCAD): Fundamentals, Simulations and Applications*. Singapore: Pan Stanford Publishing Pte. Ltd., 2017. ISBN: 978-981-4745-52-9.

[17] D. Hisamoto et al. 'FinFET-a self-aligned double-gate MOSFET scalable to 20 nm'. In: *IEEE Transactions on Electron Devices* 47.12 (2000), pp. 2320–2325. DOI: 10.1109/16.887014.

[18] James F Gibbons. 'Ion implantation in semiconductors–Part II: Damage production and annealing'. In: *Proceedings of the IEEE* 60.9 (1972), pp. 1062–1096.

[19] Avirup Dasgupta et al. 'Compact Model for Geometry Dependent Mobility in Nanosheet FETs'. In: *IEEE Electron Device Letters* 41.3 (2020), pp. 313–316. DOI: 10.1109/led.2020.2967782.

[20] J. P. Duarte et al. 'Negative-Capacitance FinFETs: Numerical Simulation, Compact Modeling and Circuit Evaluation'. In: *2018 International Conference on Simulation of Semiconductor Processes and Devices (SISPAD)*. 2018, pp. 123–128. DOI: 10.1109/sispad.2018.8551641.

[21] Hossain M. Fahad, Chenming Hu and Muhammad M. Hussain. 'Simulation Study of a 3-D Device Integrating FinFET and UTBFET'. In: *IEEE Transactions on Electron Devices* 62.1 (2015), pp. 83–87. DOI: 10.1109/ted.2014.2372695.

[22] Yupeng Chang et al. 'A survey on evaluation of large language models'. In: *ACM Transactions on Intelligent Systems and Technology* 15.3 (2024), pp. 1–45.

[23] Mahboob Elahi et al. 'A Comprehensive Literature Review of the Applications of AI Techniques through the Lifecycle of Industrial Equipment'. In: *Discover Artificial Intelligence* 3.1 (Dec. 2023), p. 43. ISSN: 2731-0809. DOI: 10.1007/s44163-023-00089-x.

[24] Ravil I. Mukhamediev et al. 'Review of Artificial Intelligence and Machine Learning Technologies: Classification, Restrictions, Opportunities and Challenges'. In: *Mathematics* 10.15 (Jan. 2022), p. 2552. ISSN: 2227-7390. DOI: 10.3390/math10152552.

[25] Ansys. *Ansys SimAI | Generative AI for Accelerated Simulation*. https://www.ansys.com/products/simai. [Online]. 2024.

[26] Stuart J. Russell et al. *Artificial Intelligence: A Modern Approach*. Fourth edition, global edition. Pearson Series in Artificial Intelligence. Harlow: Pearson, 2022. ISBN: 978-1-292-40113-3.

[27] Yann LeCun, Yoshua Bengio and Geoffrey Hinton. 'Deep Learning'. In: *Nature* 521.7553 (May 2015), pp. 436–444. ISSN: 1476-4687. DOI: 10.1038/nature14539.

[28] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. New York: Springer, 2006. ISBN: 978-0-387-31073-2.

[29] Guyue Huang et al. *Machine Learning for Electronic Design Automation: A Survey*. Mar. 2021. arXiv: 2102.03357 [cs, eess].

[30] Winston Haaswijk et al. 'Deep learning for logic optimization algorithms'. In: *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*. Ieee. 2018, pp. 1–4.

[31] Abdelrahman Hosny et al. 'DRiLLS: Deep reinforcement learning for logic synthesis'. In: *2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC)*. Ieee. 2020, pp. 581–586.

[32] Azalia Mirhoseini et al. 'Chip placement with deep reinforcement learning'. In: *arXiv preprint arXiv:2004.10746* (2020).

[33] Zhiyao Xie et al. 'Routenet: Routability prediction for mixed-size designs using convolutional neural network. In 2018 IEEE'. In: *ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 1–8.

[34] Yuzhe Ma et al. 'High performance graph convolutional networks with applications in testability analysis'. In: *Proceedings of the 56th Annual Design Automation Conference 2019*. 2019, pp. 1–6.

[35] Biruk Mammo et al. 'BugMD: Automatic mismatch diagnosis for bug triaging'. In: *2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. Ieee. 2016, pp. 1–7.

[36] Engin Afacan et al. 'Review: Machine Learning Techniques in Analog/RF Integrated Circuit Design, Synthesis, Layout, and Test'. In: *Integration* 77 (Mar. 2021), pp. 113–130. ISSN: 01679260. DOI: 10.1016/j.vlsi.2020.11.006.

[37] Zhenxin Zhao and Lihong Zhang. 'Deep reinforcement learning for analog circuit sizing'. In: *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*. Ieee. 2020, pp. 1–5.

[38] Keertana Settaluri et al. *AutoCkt: Deep Reinforcement Learning of Analog Circuit Designs*. Jan. 2020. arXiv: 2001.01808 [eess].

[39] Daniel Guerra et al. 'Artificial neural networks as an alternative for automatic analog IC placement'. In: *2019 16th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*. Ieee. 2019, pp. 1–4.

[40] Biying Xu et al. 'Wellgan: Generative-adversarial-network-guided well generation for analog/mixed-signal circuit layout'. In: *Proceedings of the 56th Annual Design Automation Conference 2019*. 2019, pp. 1–6.

[41] Feng Li and Peng-Yung Woo. 'Fault detection for linear analog IC-the method of short-circuit admittance parameters'. In: *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* 49.1 (2002), pp. 105–108.

[42] P. Burrascano, S. Fiori and M. Mongiardo. 'A Review of Artificial Neural Networks Applications in Microwave Computer-Aided Design (Invited Article)'. In: *International Journal of RF and Microwave Computer-Aided Engineering* 9.3 (1999), pp. 158–174. ISSN: 1099-047x. DOI: 10.1002/(sici)1099-047x(199905)9:3<158::aid-mmce3>3.0.co;2-v.

[43] Qi-Jun Zhang, K.C. Gupta and V.K. Devabhaktuni. 'Artificial Neural Networks for Rf and Microwave Design-from Theory to Practice'. In: *IEEE Transactions on Microwave Theory and Techniques* 51.4 (Apr. 2003), pp. 1339–1350. ISSN: 0018-9480. DOI: 10.1109/tmtt.2003.809179.

[44] Feng Feng et al. 'Artificial Neural Networks for Microwave Computer-Aided Design: The State of the Art'. In: *IEEE Transactions on Microwave Theory and Techniques* 70.11 (Nov. 2022), pp. 4597–4619. ISSN: 1557-9670. DOI: 10.1109/tmtt.2022.3197751.

[45] T-S Horng, C-C Wang and Nicolaos G Alexopoulos. 'Microstrip circuit design using neural networks'. In: *1993 IEEE MTT-S International Microwave Symposium Digest*. Ieee. 1993, pp. 413–416.

[46] P Watson and KC Gupta. 'EM-ANN models for via interconnects in microstrip circuits'. In: *1996 IEEE MTT-S International Microwave Symposium Digest*. Vol. 3. Ieee. 1996, pp. 1819–1822.

[47] GL Creech et al. 'Artificial neural networks for accurate microwave CAD applications'. In: *1996 IEEE MTT-S International Microwave Symposium Digest*. Vol. 2. Ieee. 1996, pp. 733–736.

[48] Florian Klemme et al. 'Modeling emerging technologies using machine learning: challenges and opportunities'. In: *Proceedings of the 39th International Conference on Computer-Aided Design*. Iccad '20. New York, NY, USA: Association for Computing Machinery, 2020. ISBN: 9781450380263. DOI: 10.1145/3400302.3415770.

[49] Rachita Ghoshhajra, Kalyan Biswas and Angsuman Sarkar. 'A Review on Machine Learning Approaches for Predicting the Effect of Device Parameters on Performance of Nanoscale MOSFETs'. In: *2021 Devices for Integrated Circuit (DevIC)*. 2021, pp. 489–493. DOI: 10.1109/DevIC50843.2021.9455840.

[50] Xufan Li et al. 'Overview of emerging semiconductor device model methodologies: From device physics to machine learning engines'. In: *Fundamental Research* (2024). ISSN: 2667-3258. DOI: https://doi.org/10.1016/j.fmre.2024.01.010.

[51] J.E. Rayas-Sanchez. 'EM-Based Optimization of Microwave Circuits Using Artificial Neural Networks: The State-of-the-Art'. In: *IEEE Transactions on Microwave Theory and Techniques* 52.1 (Jan. 2004), pp. 420–435. ISSN: 0018-9480. DOI: 10.1109/tmtt.2003.820897.

[52] Bo Liu, Georges Gielen and Francisco V. Fernández. *Automated Design of Analog and High-Frequency Circuits: A Computational Intelligence Approach*. Vol. 501. Studies in Computational Intelligence. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014. ISBN: 978-3-642-39161-3. DOI: 10.1007/978-3-642-39162-0.

[53] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. 2nd ed. Springer Series in Operations Research. New York: Springer, 2006. ISBN: 978-0-387-30303-1.

[54] Sébastien Bubeck. 'Convex Optimization: Algorithms and Complexity'. In: *Foundations and Trends® in Machine Learning* 8.3-4 (2015), pp. 231–357. ISSN: 1935-8237, 1935-8245. DOI: 10.1561/2200000050.

[55] Donald R Jones, Matthias Schonlau and William J Welch. 'Efficient global optimization of expensive black-box functions'. In: *Journal of Global optimization* 13 (1998), pp. 455–492.

[56] Mohammad Nabi Omidvar, Xiaodong Li and Xin Yao. 'A Review of Population-Based Metaheuristics for Large-Scale Black-Box Global Optimization—Part I'. In: *IEEE Transactions on Evolutionary Computation* 26.5 (Oct. 2022), pp. 802–822. ISSN: 1941-0026. DOI: 10.1109/tevc.2021.3130838.

[57] Jian-Yu Li, Zhi-Hui Zhan and Jun Zhang. 'Evolutionary Computation for Expensive Optimization: A Survey'. In: *Machine Intelligence Research* 19.1 (Feb. 2022), pp. 3–23. ISSN: 2731-538x, 2731-5398. DOI: 10.1007/s11633-022-1317-4.

[58] Xiwen Cai, Liang Gao and Xinyu Li. 'Efficient Generalized Surrogate-Assisted Evolutionary Algorithm for High-Dimensional Expensive Problems'. In: *IEEE Transactions on Evolutionary Computation* 24.2 (Apr. 2020), pp. 365–379. ISSN: 1941-0026. DOI: 10.1109/tevc.2019.2919762.

[59] Michael James David Powell. *Approximation theory and methods*. Cambridge university press, 1981.

[60] Jeffrey C Lagarias et al. 'Convergence properties of the Nelder–Mead simplex method in low dimensions'. In: *SIAM Journal on optimization* 9.1 (1998), pp. 112–147.

[61] *Optimizing Nonlinear Functions - MATLAB & Simulink - MathWorks United Kingdom*. https://uk.mathworks.com/help/matlab/math/optimizing-nonlinear-functions.html#bsgpq6p-11. [Online].

[62] Rainer Storn and Kenneth Price. 'Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces'. In: *International computer science institute* (1995).

[63] Kenneth Price, Rainer M Storn and Jouni A Lampinen. *Differential evolution: a practical approach to global optimization*. Springer Science & Business Media, 2006.

[64] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. 3. print. Adaptive Computation and Machine Learning. Cambridge, Mass.: MIT Press, 2008. ISBN: 978-0-262-18253-9.

[65] Jie Wang. 'An Intuitive Tutorial to Gaussian Process Regression'. In: *Computing in Science & Engineering* 25.4 (July 2023), pp. 4–11. ISSN: 1521-9615, 1558-366x. DOI: 10.1109/mcse.2023.3342149. arXiv: 2009.10862 [cs, stat].

[66] Warren S McCulloch and Walter Pitts. 'A logical calculus of the ideas immanent in nervous activity'. In: *The bulletin of mathematical biophysics* 5 (1943), pp. 115–133.

[67] Jimmy Lei Ba, Jamie Ryan Kiros and Geoffrey E. Hinton. *Layer Normalization*. July 2016. DOI: 10.48550/arXiv.1607.06450. arXiv: 1607.06450 [cs, stat].

[68] Sergey Ioffe and Christian Szegedy. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. Mar. 2015. arXiv: 1502.03167 [cs].

[69] David E. Rumelhart, Geoffrey E. Hinton and Ronald J. Williams. 'Learning Representations by Back-Propagating Errors'. In: *Nature* 323.6088 (Oct. 1986), pp. 533–536. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/323533a0.

[70] Diederik P Kingma. 'Adam: A method for stochastic optimization'. In: *arXiv preprint arXiv:1412.6980* (2014).

[71] Kurt Hornik. 'Approximation capabilities of multilayer feedforward networks'. In: *Neural networks* 4.2 (1991), pp. 251–257.

[72] Charles Blundell et al. *Weight Uncertainty in Neural Networks*. May 2015. arXiv: 1505.05424 [cs, stat].

[73] Kevin P. Murphy. *Probabilistic Machine Learning: An Introduction*. Adaptive Computation and Machine Learning. Cambridge, Massachusetts London, England: The MIT Press, 2022. ISBN: 978-0-262-04682-4.

[74] Jonas Močkus. 'Bayesian approach to global optimization'. In: *Mathematics and its applications. Kluwer Academic* (1989).

[75] Jonas Močkus. 'On Bayesian methods for seeking the extremum'. In: *Optimization techniques IFIP technical conference: Novosibirsk, July 1–7, 1974*. Springer. 1975, pp. 400–404.

[76] Michael TM Emmerich, Kyriakos C Giannakoglou and Boris Naujoks. 'Single- and multiobjective evolutionary optimization assisted by Gaussian random field metamodels'. In: *IEEE Transactions on Evolutionary Computation* 10.4 (2006), pp. 421–439.

[77] David M. Pozar. *Microwave and RF Wireless Systems*. New York Weinheim: Wiley, 2001. ISBN: 978-0-471-32282-5.

[78] Richard J. Cameron, Chandra M. Kudsia and Raafat R. Mansour. *Microwave Filters for Communication Systems: Fundamentals, Design and Applications*. 2nd ed. Hoboken: Wiley, 2018. ISBN: 978-1-119-29239-5.

[79] Giuseppe Macchiarella and Stefano Tamiazzo. 'Cooking Microwave Filters: Is Synthesis Still Helpful in Microwave Filter Design?' In: *IEEE Microwave Magazine* 21.3 (Mar. 2020), pp. 20–33. ISSN: 1527-3342, 1557-9581. DOI: 10.1109/mmm.2019.2958148.

[80] Richard J. Cameron, Chandra M. Kudsia and Raafat R. Mansour. 'Synthesis of a General Class of the Chebyshev Filter Function'. In: *Microwave Filters for Communication Systems: Fundamentals, Design, and Applications*. 2018, pp. 177–213. DOI: 10.1002/9781119292371.ch6.

[81] R.J. Cameron. 'Advanced coupling matrix synthesis techniques for microwave filters'. In: *IEEE Transactions on Microwave Theory and Techniques* 51.1 (2003), pp. 1–10. DOI: 10.1109/tmtt.2002.806937.

[82] G. Macchiarella. 'Accurate synthesis of inline prototype filters using cascaded triplet and quadruplet sections'. In: *IEEE Transactions on Microwave Theory and Techniques* 50.7 (2002), pp. 1779–1783. DOI: 10.1109/tmtt.2002.800429.

[83] S. Amari. 'Synthesis of cross-coupled resonator filters using an analytical gradient-based optimization technique'. In: *IEEE Transactions on Microwave Theory and Techniques* 48.9 (2000), pp. 1559–1564. DOI: 10.1109/22.869008.

[84] Bo Liu, Hao Yang and Michael J. Lancaster. 'Synthesis of Coupling Matrix for Diplexers Based on a Self-Adaptive Differential Evolution Algorithm'. In: *IEEE Transactions on Microwave Theory and Techniques* 66.2 (2018), pp. 813–821. DOI: 10.1109/tmtt.2017.2772855.

[85] Xun Luo, Bingzheng Yang and Huizhen Jenny Qian. 'Adaptive Synthesis for Resonator-Coupled Filters Based on Particle Swarm Optimization'. In: *IEEE Transactions on Microwave Theory and Techniques* 67.2 (2019), pp. 712–725. DOI: 10.1109/tmtt.2018.2878197.

[86] Sanghoon Shin and Sridhar Kanamaluru. 'Diplexer design using EM and circuit simulation techniques'. In: *IEEE Microwave Magazine* 8.2 (2007), pp. 77–82. DOI: 10.1109/mmw.2007.335532.

[87] Jia-Shen G Hong and Michael J Lancaster. *Microstrip filters for RF/microwave applications*. John Wiley & Sons, 2004.

[88] Matthias Caenepeel et al. 'Parametric modeling of the coupling parameters of planar coupled-resonator microwave filters'. In: *2015 European Microwave Conference (EuMC)*. Ieee. 2015, pp. 538–541.

[89] Matthias Caenepeel, Francesco Ferranti and Yves Rolain. 'Efficient and automated generation of multidimensional design curves for coupled-resonator filters using system identification and metamodels'. In: *2016 13th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*. Ieee. 2016, pp. 1–4.

[90] Yang Yu et al. 'State-of-the-Art: AI-Assisted Surrogate Modeling and Optimization for Microwave Filters'. In: *IEEE Transactions on Microwave Theory and Techniques* 70.11 (Nov. 2022), pp. 4635–4651. ISSN: 1557-9670. DOI: 10.1109/tmtt.2022.3208898.

[91] M. Guglielmi. 'Simple CAD procedure for microwave filters and multiplexers'. In: *IEEE Transactions on Microwave Theory and Techniques* 42.7 (1994), pp. 1347–1352. DOI: 10.1109/22.299728.

[92] Xiaobang Shang, Wenlin Xia and Michael J Lancaster. 'The design of waveguide filters based on cross-coupled resonators'. In: *Microwave and Optical Technology Letters* 56.1 (2014), pp. 3–8.

[93] Daniel Swanson. 'Narrow-band microwave filter design'. In: *IEEE Microwave Magazine* 8 (2007), pp. 105–114.

[94] José E. Rayas-Sanchez, Slawomir Koziel and John W. Bandler. 'Advanced RF and Microwave Design Optimization: A Journey and a Vision of Future Trends'. In: *IEEE Journal of Microwaves* 1.1 (2021), pp. 481–493. DOI: 10.1109/jmw.2020.3034263.

[95] J.W. Bandler et al. 'Space Mapping: The State of the Art'. In: *IEEE Transactions on Microwave Theory and Techniques* 52.1 (Jan. 2004), pp. 337–361. ISSN: 0018-9480. DOI: 10.1109/tmtt.2003.820904.

[96] S. Koziel, J.W. Bandler and K. Madsen. 'A Space-Mapping Framework for Engineering Optimization–Theory and Implementation'. In: *IEEE Transactions on Microwave Theory and Techniques* 54.10 (2006), pp. 3721–3730. DOI: 10.1109/tmtt.2006.882894.

[97] Chao Zhang et al. 'Cognition-Driven Formulation of Space Mapping for Equal-Ripple Optimization of Microwave Filters'. In: *IEEE Transactions on Microwave Theory and Techniques* 63.7 (July 2015), pp. 2154–2165. ISSN: 0018-9480, 1557-9670. DOI: 10.1109/tmtt.2015.2431675.

[98] Feng Feng et al. 'Multifeature-Assisted Neuro-Transfer Function Surrogate-Based EM Optimization Exploiting Trust-Region Algorithms for Microwave Filter Design'. In: *IEEE Transactions on Microwave Theory and Techniques* 68.2 (Feb. 2020), pp. 531–542. ISSN: 0018-9480, 1557-9670. DOI: 10.1109/tmtt.2019.2952101.

[99] Ping Zhao and Ke Wu. 'Homotopy Optimization of Microwave and Millimeter-Wave Filters Based on Neural Network Model'. In: *IEEE Transactions on Microwave Theory and Techniques* 68.4 (Apr. 2020), pp. 1390–1400. ISSN: 0018-9480, 1557-9670. DOI: 10.1109/tmtt.2019.2963639.

[100] Seymour Cohn. 'Direct-Coupled-Resonator Filters'. In: *Proceedings of the IRE* 45.2 (1957), pp. 187–196. ISSN: 0096-8390. DOI: 10.1109/jrproc.1957.278389.

[101] J.W. Bandler et al. 'Space Mapping Technique for Electromagnetic Optimization'. In: *IEEE Transactions on Microwave Theory and Techniques* 42.12 (Dec. 1994), pp. 2536–2544. ISSN: 00189480. DOI: 10.1109/22.339794.

[102] Jose E. Rayas-Sanchez. 'Power in Simplicity with ASM: Tracing the Aggressive Space Mapping Algorithm Over Two Decades of Development and Engineering Applications'. In: *IEEE Microwave Magazine* 17.4 (2016), pp. 64–76. DOI: 10.1109/mmm.2015.2514188.

[103] J.W. Bandler et al. 'Implicit space mapping optimization exploiting preassigned parameters'. In: *IEEE Transactions on Microwave Theory and Techniques* 52.1 (2004), pp. 378–385. DOI: 10.1109/tmtt.2003.820892.

[104] M.H. Bakr et al. 'Neural space-mapping optimization for EM-based design'. In: *IEEE Transactions on Microwave Theory and Techniques* 48.12 (2000), pp. 2307–2315. DOI: 10.1109/22.898979.

[105] J.W. Bandler et al. 'Neural inverse space mapping EM-optimization'. In: *2001 IEEE MTT-S International Microwave Sympsoium Digest (Cat. No.01CH37157)*. Vol. 2. 2001, 1007–1010 vol.2. DOI: 10.1109/mwsym.2001.967062.

[106] Qingsha S. Cheng et al. 'The State of the Art of Microwave CAD: EM-Based Optimization and Modeling: The State of the Art of Microwave CAD'. In: *International Journal of RF and Microwave Computer-Aided Engineering* 20.5 (Sept. 2010), pp. 475–491. ISSN: 10964290. DOI: 10.1002/mmce.20454.

[107] Mohamed Yahia et al. 'Ridged waveguide filter optimization using an improved simplex method'. In: *2012 6th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)*. Ieee. 2012, pp. 203–206.

[108] Feng Feng et al. 'Parallel EM Optimization Approach to Microwave Filter Design Using Feature Assisted Neuro-Transfer Functions'. In: *2016 IEEE MTT-S International Microwave Symposium (IMS)*. San Francisco, CA: Ieee, May 2016, pp. 1–3. ISBN: 978-1-5090-0698-4. DOI: 10.1109/mwsym.2016.7539963.

[109] Milad Sharifi Sorkherizi and Ahmed A. Kishk. 'Use of Group Delay of Sub-Circuits in Optimization of Wideband Large-Scale Bandpass Filters and Diplexers'. In: *IEEE Transactions on Microwave Theory and Techniques* 65.8 (Aug. 2017), pp. 2893–2905. ISSN: 0018-9480, 1557-9670. DOI: 10.1109/tmtt.2017.2669969.

[110] Bo Liu, Hao Yang and Michael J. Lancaster. 'Global Optimization of Microwave Filters Based on a Surrogate Model-Assisted Evolutionary Algorithm'. In: *IEEE Transactions on Microwave Theory and Techniques* 65.6 (June 2017), pp. 1976–1985. ISSN: 0018-9480, 1557-9670. DOI: 10.1109/tmtt.2017.2661739.

[111] Pei-Wen Shu, Qing-Xin Chu and Jian-Ye Mai. 'Harris Hawks Optimization Algorithm for Waveguide Filter Designs'. In: *2020 IEEE Asia-Pacific Microwave Conference (APMC)*. Hong Kong, Hong Kong: Ieee, Dec. 2020, pp. 406–408. ISBN: 978-1-72816-962-0. DOI: 10.1109/apmc47863.2020.9331450.

[112] Yanxing Wang et al. 'Accurate microwave filter design based on particle swarm optimization and one-dimensional convolution autoencoders'. In: *International Journal of RF and Microwave Computer-Aided Engineering* 32.4 (2022), e23034.

[113] D.H. Wolpert and W.G. Macready. 'No free lunch theorems for optimization'. In: *IEEE Transactions on Evolutionary Computation* 1.1 (1997), pp. 67–82. DOI: 10.1109/4235.585893.

[114] Yang Yu. 'Knowledge-Guided Microwave Diplexers and Multiplexers Design based on Computational Intelligent Techniques'. PhD thesis. University of Birmingham, Feb. 2022.

[115] Douglas C Montgomery. *Design and analysis of experiments*. John wiley & sons, 2017.

[116] Nicholas Metropolis and Stanislaw Ulam. 'The monte carlo method'. In: *Journal of the American statistical association* 44.247 (1949), pp. 335–341.

[117] Michael D McKay, Richard J Beckman and William J Conover. 'A comparison of three methods for selecting values of input variables in the analysis of output from a computer code'. In: *Technometrics* 42.1 (2000), pp. 55–61.

[118] David M. Pozar. *Microwave Engineering*. Fourth edition. Hoboken, NJ: John Wiley & Sons, Inc, 2012. ISBN: 978-0-470-63155-3.

[119] Dirk Deschrijver and Tom Dhaene. 'A Note on the Multiplicity of Poles in the Vector Fitting Macromodeling Method'. In: *IEEE Transactions on Microwave Theory and Techniques* 55.4 (2007), pp. 736–741. DOI: 10.1109/tmtt.2007.893651.

[120] Ping Zhao and Ke-Li Wu. 'Circuit Model Extraction of Parallel-Connected Dual-Passband Coupled-Resonator Filters'. In: *IEEE Transactions on Microwave Theory and Techniques* 66.2 (2018), pp. 822–830. DOI: 10.1109/tmtt.2017.2764086.

[121] B. Gustavsen and A. Semlyen. 'Rational approximation of frequency domain responses by vector fitting'. In: *IEEE Transactions on Power Delivery* 14.3 (1999), pp. 1052–1061. DOI: 10.1109/61.772353.

[122] Bo Liu, Vic Grout and Anna Nikolaeva. 'Efficient global optimization of actuator based on a surrogate model assisted hybrid algorithm'. In: *IEEE Transactions on Industrial Electronics* 65.7 (2017), pp. 5712–5721.

[123] Xiaobang Shang et al. 'Design of multiple-passband filters using coupling matrix optimisation'. In: *IET microwaves, antennas & propagation* 6.1 (2012), pp. 24–30.

[124] E. Ofli, R. Vahldieck and S. Amari. 'Novel E-plane filters and diplexers with elliptic response for millimeter-wave applications'. In: *IEEE Transactions on Microwave Theory and Techniques* 53.3 (2005), pp. 843–851. DOI: 10.1109/tmtt.2004.842506.

[125] Elisa Cipriani et al. 'Theoretical and experimental comparison of Class F vs. Class F- 1 PAs'. In: *The 5th European Microwave Integrated Circuits Conference*. Ieee. 2010, pp. 428–431.

[126] Bumman Kim, Ildu Kim and Junghwan Moon. 'Advanced doherty architecture'. In: *IEEE Microwave Magazine* 11.5 (2010), pp. 72–86.

[127] Frederick Raab. 'Efficiency of outphasing RF power-amplifier systems'. In: *IEEE Transactions on communications* 33.10 (1985), pp. 1094–1099.

[128] Lida Kouhalvandi, Osman Ceylan and Serdar Ozoguz. 'A Review on Optimization Methods for Designing RF Power Amplifiers'. In: *2019 11th International Conference on Electrical and Electronics Engineering (ELECO)*. Nov. 2019, pp. 375–378. DOI: 10.23919/eleco47770.2019.8990396.

[129] Chuan Li et al. 'Simulated Annealing Particle Swarm Optimization for High-Efficiency Power Amplifier Design'. In: *IEEE Transactions on Microwave Theory and Techniques* 69.5 (May 2021), pp. 2494–2505. ISSN: 0018-9480, 1557-9670. DOI: 10.1109/tmtt.2021.3061547.

[130] Han Liu et al. 'Simulated Annealing Particle Swarm Optimization for a Dual-Input Broadband GaN Doherty Like Load-Modulated Balance Amplifier Design'. In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 69.9 (Sept. 2022), pp. 3734–3738. ISSN: 1549-7747, 1558-3791. DOI: 10.1109/tcsii.2022.3173608.

[131] Lida Kouhalvandi, Osman Ceylan and Serdar Ozoguz. 'Automated Deep Neural Learning-Based Optimization for High Performance High Power Amplifier Designs'. In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 67.12 (Dec. 2020), pp. 4420–4433. ISSN: 1549-8328, 1558-0806. DOI: 10.1109/tcsi.2020.3008947.

[132] Engin Afacan et al. 'Machine learning techniques in analog/RF integrated circuit design, synthesis, layout, and test'. In: *Integration* 77 (2021), pp. 113–130.

[133] Fábio Passos et al. 'A 23.5–32.5 GHz, 17dBm P SAT and 37.5% PAE Power Amplifier Synthesized Using an Automated Design Methodology'. In: *2023 19th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*. Ieee. 2023, pp. 1–4.

[134] Peng Chen, Brian M. Merrick and Thomas J. Brazil. 'Bayesian Optimization for Broadband High-Efficiency Power Amplifier Designs'. In: *IEEE Transactions on Microwave Theory and Techniques* 63.12 (Dec. 2015), pp. 4263–4272. ISSN: 0018-9480, 1557-9670. DOI: 10.1109/tmtt.2015.2495360.

[135] Peng Chen et al. 'Multiobjective Bayesian Optimization for Active Load Modulation in a Broadband 20-W GaN Doherty Power Amplifier Design'. In: *IEEE Transactions on Microwave Theory and Techniques* 65.3 (Mar. 2017), pp. 860–871. ISSN: 0018-9480, 1557-9670. DOI: 10.1109/tmtt.2016.2636146.

[136] Jia Guo, Giovanni Crupi and Jialin Cai. 'A Broadband Asymmetric Doherty Power Amplifier Design Based on Multiobjective Bayesian Optimization: Theoretical and Experimental Validation'. In: *IEEE Access* 10 (2022), pp. 89823–89834. ISSN: 2169-3536. DOI: 10.1109/access.2022.3201348.

[137] Ramazan Kopru, Hakan Kuntman and B. S. Yarman. 'A Novel Method to Design Wideband Power Amplifier for Wireless Communication'. In: *2013 IEEE International Symposium on Circuits and Systems (ISCAS2013)*. Beijing: Ieee, May 2013, pp. 1942–1945. ISBN: 978-1-4673-5762-3. DOI: 10.1109/iscas.2013.6572248.

[138] Ramazan Köprü, Hakan Kuntman and Binboga Siddik Yarman. 'On Numerical Design Technique of Wideband Microwave Amplifiers Based on GaN Small-Signal Device Model'. In: *Analog Integrated Circuits and Signal Processing* 81.1 (Oct. 2014), pp. 71–87. ISSN: 0925-1030, 1573-1979. DOI: 10.1007/s10470-014-0355-4.

[139] Peng Chen, Songbai He and Fei You. 'Automated Power Amplifier Design Assisted with Particle Swarm Optimization'. In: *2012 Asia Pacific Microwave Conference Proceedings*. Kaohsiung, Taiwan: Ieee, Dec. 2012, pp. 481–483. ISBN: 978-1-4577-1332-3. DOI: 10.1109/apmc.2012.6421637.

[140] A. A. Kokolov et al. 'Design of Harmonic-Tuned Dual-Band GaN HEMT Power Amplifier Based on Genetic Algorithm'. In: *2014 24th International Crimean Conference Microwave & Telecommunication Technology*. Sevastopol, Ukraine: Ieee, Sept. 2014, pp. 95–96. ISBN: 978-966-335-417-0. DOI: 10.1109/crmico.2014.6959305.

[141] Eyad Arabi et al. 'Design of a Triple-Band Power Amplifier Using a Genetic Algorithm and the Continuous Mode Method'. In: *2017 IEEE Topical Conference on RF/Microwave Power Amplifiers for Radio and Wireless Applications (PAWR)*. Phoenix, AZ, USA: Ieee, Jan. 2017, pp. 48–51. ISBN: 978-1-5090-3458-1. DOI: 10.1109/pawr.2017.7875570.

[142] S. Ali Hosseini, Ahmad Hajipour and Hamidreza Tavakoli. 'Design and Optimization of a CMOS Power Amplifier Using Innovative Fractional-Order Particle Swarm Optimization'. In: *Applied Soft Computing* 85 (Dec. 2019), p. 105831. ISSN: 15684946. DOI: 10.1016/j.asoc.2019.105831.

[143] Bingjie Sun et al. 'A Wide-Band Power Amplifier Based on Genetic Algorithm and Direct Layout Optimization'. In: *2022 IEEE MTT-S International Microwave Workshop Series on Advanced Materials and Processes for RF and THz Applications (IMWS-AMP)*. Guangzhou, China: Ieee, Nov. 2022, pp. 1–3. ISBN: 978-1-66547-834-2. DOI: 10.1109/imws-amp54652.2022.10107103.

[144] Catarina Belchior et al. 'Automatic Methodology for Wideband Power Amplifier Design'. In: *IEEE Microwave and Wireless Components Letters* 31.8 (Aug. 2021), pp. 989–992. ISSN: 1531-1309, 1558-1764. DOI: 10.1109/lmwc.2021.3083101.

[145] Catarina Belchior et al. 'Towards the Automated RF Power Amplifier Design'. In: *2023 19th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*. July 2023, pp. 1–4. DOI: 10.1109/smacd58065.2023.10192148.

[146] Bo Liu et al. 'GASPAD: A General and Efficient Mm-Wave Integrated Circuit Synthesis Method Based on Surrogate Model Assisted Evolutionary Algorithm'. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 33.2 (Feb. 2014), pp. 169–182. ISSN: 0278-0070, 1937-4151. DOI: 10.1109/tcad.2013.2284109.

[147] Nicolas Knudde et al. 'Data-Efficient Bayesian Optimization with Constraints for Power Amplifier Design'. In: *2018 IEEE MTT-S International Conference on Numerical Electromagnetic and Multiphysics Modeling and Optimization (NEMO)*. Reykjavik: Ieee, Aug. 2018, pp. 1–3. ISBN: 978-1-5386-5204-6. DOI: 10.1109/nemo.2018.8503107.

[148] Jia Guo, Giovanni Crupi and Jialin Cai. 'A Novel Design Methodology for a Multioctave GaN-HEMT Power Amplifier Using Clustering Guided Bayesian Optimization'. In: *IEEE Access* 10 (2022), pp. 52771–52781. ISSN: 2169-3536. DOI: 10.1109/access.2022.3175870.

[149] Lida Kouhalvandi, Osman Ceylan and Serdar Ozoguz. 'Automated RF Power Amplifier Optimization and Design: From Lumped Elements to Distributed Elements'. In: *2019 27th Telecommunications Forum (TELFOR)*. Belgrade, Serbia: Ieee, Nov. 2019, pp. 1–4. ISBN: 978-1-72814-790-1. DOI: 10.1109/telfor48224.2019.8971160.

[150] Yinshuang Zhao et al. 'Multiobjective Bayesian Optimization for a 1.7-3.7GHz GaN Power Amplifier Design'. In: *2022 IEEE MTT-S International Wireless Symposium (IWS)*. Harbin, China: Ieee, Aug. 2022, pp. 1–3. ISBN: 978-1-66548-197-7. DOI: 10.1109/iws55252.2022.9977507.

[151] Steve C Cripps et al. *RF power amplifiers for wireless communications*. Vol. 250. Artech house Norwood, MA, 2006.

[152] BS Yarman and HJ Carlin. 'A simplified" real frequency" technique applied to broad-band multistage microwave amplifiers'. In: *IEEE Transactions on Microwave Theory and Techniques* 30.12 (1982), pp. 2216–2222.

[153] Steve Marsh, ed. *Practical MMIC Design*. Artech House Microwave Library. Norwood, MA: Artech House, 2011. ISBN: 978-1-59693-036-0.

[154] Rocco Giofrè, Alessandro Del Gaudio and Ernesto Limiti. 'A 28 GHz MMIC Doherty Power Amplifier in GaN on Si Technology for 5G Applications'. In: *2019 IEEE MTT-S International Microwave Symposium (IMS)*. 2019, pp. 611–613. DOI: 10.1109/mwsym.2019.8700757.

[155] Mingquan Bao et al. 'A 24-–28-GHz Doherty Power Amplifier With 4-W Output Power and 32% PAE at 6-dB OPBO in 150-nm GaN Technology'. In: *IEEE Microwave and Wireless Components Letters* 31.6 (2021), pp. 752–755. DOI: 10.1109/lmwc.2021.3063868.

[156] Anna Piacibello et al. 'A 5-W GaN Doherty Amplifier for Ka-Band Satellite Downlink With 4-GHz Bandwidth and 17-dB NPR'. In: *IEEE Microwave and Wireless Components Letters* 32.8 (2022), pp. 964–967. DOI: 10.1109/lmwc.2022.3160227.

[157] Rui-Jia Liu et al. 'A 24-–28-GHz GaN MMIC Synchronous Doherty Power Amplifier With Enhanced Load Modulation for 5G mm-Wave Applications'. In: *IEEE Transactions on Microwave Theory and Techniques* 70.8 (2022), pp. 3910–3922. DOI: 10.1109/tmtt.2022.3176818.

[158] Anna Piacibello et al. '3-Way Doherty Power Amplifiers: Design Guidelines and MMIC Implementation at 28 GHz'. In: *IEEE Transactions on Microwave Theory and Techniques* 71.5 (2023), pp. 2016–2028. DOI: 10.1109/tmtt.2022.3225316.

[159] Peng Chen et al. 'Simplified Emulation of Active Load Modulation for a Millimeter-Wave GaN MMIC Doherty Power Amplifier Design'. In: *IEEE Transactions on Microwave Theory and Techniques* 72.1 (2024), pp. 149–159. DOI: 10.1109/tmtt.2023.3284258.

[160] Hiroshi Iwai and Durga Misra. 'The Transistor was Invented 75 Years Ago: A Big Milestone in Human History'. In: *The Electrochemical Society Interface* 31.4 (Dec. 2022), p. 65. DOI: 10.1149/2.f13224if.

[161] Chris A. Mack. 'Fifty Years of Moore's Law'. In: *IEEE Transactions on Semiconductor Manufacturing* 24.2 (2011), pp. 202–207. DOI: 10.1109/tsm.2010.2096437.

[162] R. Kalaivani et al. 'Design and Simulation of 22nm FinFET Structure Using TCAD'. In: *2020 5th International Conference on Devices, Circuits and Systems (ICDCS)*. 2020, pp. 286–289. DOI: 10.1109/icdcs48716.2020.243600.

[163] Jiahao Xie et al. 'Designing Semiconductor Materials and Devices in the Post-Moore Era by Tackling Computational Challenges with Data-Driven Strategies'. In: *Nature Computational Science* 4.5 (May 2024), pp. 322–333. ISSN: 2662-8457. DOI: 10.1038/s43588-024-00632-5.

[164] F. Benistant et al. 'TCAD Modeling for next generation CMOS devices'. In: *2014 20th International Conference on Ion Implantation Technology (IIT)*. 2014, pp. 1–6. DOI: 10.1109/iit.2014.6939997.

[165] S. Williams and K. Varahramyan. 'A new TCAD-based statistical methodology for the optimization and sensitivity analysis of semiconductor technologies'. In: *IEEE Transactions on Semiconductor Manufacturing* 13.2 (2000), pp. 208–218. DOI: 10.1109/66.843636.

[166] Jing Wang et al. 'Artificial Neural Network-Based Compact Modeling Methodology for Advanced Transistors'. en. In: *IEEE Transactions on Electron Devices* 68.3 (Mar. 2021), pp. 1318–1325. ISSN: 0018-9383, 1557-9646. DOI: 10.1109/ted.2020.3048918.

[167] Anh Nguyen et al. 'Fully analog ReRAM neuromorphic circuit optimization using DTCO simulation framework'. In: *International Conference on Simulation of Semiconductor Processes and Devices, SISPAD* 2020-September (Sept. 2020), pp. 201–204. DOI: 10.23919/sispad49475.2020.9241635.

[168] Jeehwan Song, Jian Ping Wang and Chris H. Kim. 'MRAM DTCO and compact models'. In: *Technical Digest - International Electron Devices Meeting, IEDM* 2020-December (Dec. 2020), pp. 41.6.1–41.6.4. DOI: 10.1109/iedm13553.2020.9371928.

[169] Y. K. Cheng et al. 'Next-generation design and technology co-optimization (DTCO) of system on integrated chip (SoIC) for mobile and HPC applications'. In: *Technical Digest - International Electron Devices Meeting, IEDM* 2020-December (Dec. 2020), pp. 41.3.1–41.3.4. DOI: 10.1109/iedm13553.2020.9372005.

[170] V. Moroz et al. 'DTCO launches moore's law over the feature scaling wall'. In: *Technical Digest - International Electron Devices Meeting, IEDM* 2020-December (Dec. 2020), pp. 41.1.1–41.1.4. DOI: 10.1109/iedm13553.2020.9372010.

[171] Qinghua Han et al. 'A DTCO approach on DRAM bit line capacitance and sensing margin improvement'. In: *2020 IEEE 15th International Conference on Solid-State and Integrated Circuit Technology, ICSICT 2020 - Proceedings* (Nov. 2020). DOI: 10.1109/icsict49897.2020.9278287.

[172] Ayse Coskun et al. 'Cross-Layer Co-Optimization of Network Design and Chiplet Placement in 2.5-D Systems'. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 39 (Dec. 2020), pp. 5183–5196. DOI: 10.1109/tcad.2020.2970019.

[173] Hiu Yung Wong et al. 'TCAD-Machine learning framework for device variation and operating temperature analysis with experimental demonstration'. In: *IEEE Journal of the Electron Devices Society* 8 (2020), pp. 992–1000. DOI: 10.1109/jeds.2020.3024669.

[174] Sanghoon Myung et al. 'Real-time TCAD: A new paradigm for TCAD in the artificial intelligence era'. In: *International Conference on Simulation of Semiconductor Processes and Devices, SISPAD* 2020-September (Sept. 2020), pp. 347–350. DOI: 10.23919/sispad49475.2020.9241622.

[175] Farid Kenarangi and Inna Partin-Vaisband. 'Leveraging Independent Double-Gate FinFET Devices for Machine Learning Classification'. In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 66 (Nov. 2019), pp. 4356–4367. DOI: 10.1109/tcsi.2019.2927441.

[176] Zhe Zhang et al. 'New-generation design-technology co-optimization (DTCO): Machine-learning assisted modeling framework'. In: *2019 Silicon Nanoelectronics Workshop, SNW 2019* (June 2019). DOI: 10.23919/snw.2019.8782897.

[177] Kawsar Haghshenas, Mona Hashemi and Tooraj Nikoubin. 'Fast and Energy-Efficient CNFET Adders With CDM and Sensitivity-Based Device-Circuit Co-Optimization'. In: *IEEE Transactions on Nanotechnology* 17 (July 2018), pp. 783–794. DOI: 10.1109/tnano.2018.2834511.

[178] Z. Stanojevic et al. 'Cell designer- A comprehensive TCAD-based framework for DTCO of standard logic cells'. In: *European Solid-State Device Research Conference* 2018-September (Oct. 2018), pp. 202–205. DOI: 10.1109/essderc.2018.8486887.

[179] Asen Asenov et al. 'TCAD based Design-Technology Co-Optimisations in advanced technology nodes'. In: *2017 International Symposium on VLSI Technology, Systems and Application, VLSI-TSA 2017* (June 2017). DOI: 10.1109/vlsi-tsa.2017.7942435.

[180] A. Sasikumar and R. Muthaiah. 'Operational amplifier circuit sizing based on NSGA-II and particle swarm optimization'. In: *2017 International Conference on Networks and Advances in Computational Technologies, NetACT 2017* (Oct. 2017), pp. 64–68. DOI: 10.1109/netact.2017.8076742.

[181] Louis Gerrer et al. 'Accurate simulation of transistor-level variability for the purposes of TCAD-based device-technology cooptimization'. In: *IEEE Transactions on Electron Devices* 62 (June 2015), pp. 1739–1745. DOI: 10.1109/ted.2015.2402440.

[182] Gage Hills et al. 'Rapid Co-Optimization of Processing and Circuit Design to Overcome Carbon Nanotube Variations'. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 34 (July 2015), pp. 1082–1095. DOI: 10.1109/tcad.2015.2415492.

[183] Xingsheng Wang et al. 'FinFET centric variability-aware compact model extraction and generation technology supporting DTCO'. In: *IEEE Transactions on Electron Devices* 62 (Oct. 2015), pp. 3139–3146. DOI: 10.1109/ted.2015.2463073.

[184] J. Ryckaert et al. 'Design Technology co-optimization for N10'. In: *Proceedings of the IEEE 2014 Custom Integrated Circuits Conference, CICC 2014* (Nov. 2014). DOI: 10.1109/cicc.2014.6946037.

[185] Bo Liu et al. 'An efficient high-frequency linear RF amplifier synthesis method based on evolutionary computation and machine learning techniques'. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 31 (2012), pp. 981–993. DOI: 10.1109/tcad.2012.2187207.

[186] Tejas Jhaveri et al. 'Co-optimization of circuits, layout and lithography for predictive technology scaling beyond gratings'. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 29 (Apr. 2010), pp. 509–527. DOI: 10.1109/tcad.2010.2042882.

[187] Jatmiko E. Suseno et al. 'Artificial intelligence techniques for SPICE optimization of MOSFET modeling'. In: *2009 Innovative Technologies in Intelligent Systems and Industrial Applications, CITISIA 2009* (2009), pp. 76–80. DOI: 10.1109/citisia.2009.5224238.

[188] Mi Chang Chang et al. 'Transistor- and circuit-design optimization for low-power CMOS'. In: *IEEE Transactions on Electron Devices* 55 (Jan. 2008), pp. 84–95. DOI: 10.1109/ted.2007.911348.

[189] Lerong Cheng et al. 'Device and architecture co-optimization for FPGA power reduction'. In: (Feb. 2008), pp. 915–920. DOI: 10.1109/dac.2005.193946.

[190] Jintae Kim et al. 'Device-circuit co-optimization for mixed-mode circuit design via geometric programming'. In: *IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers, ICCAD* (2007), pp. 470–475. DOI: 10.1109/iccad.2007.4397309.

[191] Rob Roy, Debashis Bhattacharya and Vamsi Boppana. 'Transistor-level optimization of digital designs with flex cells'. In: *Computer* 38 (Feb. 2005), pp. 53–61. DOI: 10.1109/mc.2005.74.

[192] Thomas Binder, Clemens Heitzinger and Siegfried Selberherr. 'A study on global and local optimization techniques for TCAD analysis tasks'. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 23 (June 2004), pp. 814–822. DOI: 10.1109/tcad.2004.828130.

[193] Michael Schröter et al. 'Physics- and process-based bipolar transistor modeling for integrated circuit design'. In: *IEEE Journal of Solid-State Circuits* 34 (Aug. 1999), pp. 1136–1149. DOI: 10.1109/4.777111.

[194] Krishna Shenai. 'Mixed-mode circuit simulation: an emerging CAD tool for the design and optimization of power semiconductor devices and circuits'. In: *IEEE Workshop on Computers in Power Electronics* (1994), pp. 1–5. DOI: 10.1109/cipe.1994.396743.

[195] Binjie Cheng et al. 'Impact of intrinsic parameter fluctuations in decanano MOSFETs on yield and functionality of SRAM cells'. In: *Solid-State Electronics* 49.5 (2005), pp. 740–746.

[196] Ramiz Salama et al. 'Terahertz Communications and Sensing Overview'. In: *2023 3rd International Conference on Advancement in Electronics & Communication Engineering (AECE)*. 2023, pp. 393–400. DOI: 10.1109/aece59614.2023.10428466.

[197] Roger Appleby and H. Bruce Wallace. 'Standoff Detection of Weapons and Contraband in the 100 GHz to 1 THz Region'. In: *IEEE Transactions on Antennas and Propagation* 55.11 (2007), pp. 2944–2956. DOI: 10.1109/tap.2007.908543.

[198] D. Jasteh et al. 'Low-THz imaging radar for outdoor applications'. In: *2015 16th International Radar Symposium (IRS)*. 2015, pp. 203–208. DOI: 10.1109/irs.2015.7226363.

[199] S.K. Kurinec. 'Junction Field Effect Transistors'. In: *Encyclopedia of Materials: Science and Technology*. Ed. by K.H. Jürgen Buschow et al. Oxford: Elsevier, 2001, pp. 4356–4361. ISBN: 978-0-08-043152-9. DOI: https://doi.org/10.1016/B0-08-043152-6/00764-6.

[200] Xiaobing Mei et al. 'First Demonstration of Amplification at 1 THz Using 25-nm InP High Electron Mobility Transistor Process'. In: *IEEE Electron Device Letters* 36.4 (2015), pp. 327–329. DOI: 10.1109/led.2015.2407193.

[201] Bo Liu et al. 'An Efficient Method for Antenna Design Optimization Based on Evolutionary Computation and Machine Learning Techniques'. In: *IEEE Transactions on Antennas and Propagation* 62.1 (Jan. 2014), pp. 7–18. ISSN: 0018-926x, 1558-2221. DOI: 10.1109/tap.2013.2283605.

[202] *The Diramics website*. https://diramics.com/wp-content/uploads/downloads/DIRAMICS-pH-100-4F20.pdf. [Online].

[203] YC Chen et al. 'Composite-channel InP HEMT for W-band power amplifiers'. In: *Conference Proceedings. Eleventh International Conference on Indium Phosphide and Related Materials (IPRM'99)(Cat. No. 99CH36362)*. Ieee. 1999, pp. 305–306.

[204] Hiroki Sugiyama et al. 'High-electron-mobility In 0.53 Ga 0.47 As/In 0.8 Ga 0.2 As composite-channel modulation-doped structures grown by metal-organic vapor-phase epitaxy'. In: *2010 22nd International Conference on Indium Phosphide and Related Materials (IPRM)*. Ieee. 2010, pp. 1–4.

[205] MD Lange et al. 'InAs/InGaAs composite-channel HEMT on InP: Tailoring InGaAs thickness for performance'. In: *2008 20th International Conference on Indium Phosphide and Related Materials*. Ieee. 2008, pp. 1–4.

[206] Behzad Razavi. *RF Microelectronics*. 2nd ed. Upper Saddle River, NJ: Prentice Hall, 2012. ISBN: 978-0-13-713473-1.

[207] Jianan Deng et al. 'A theoretical study of gating effect on InP-InGaAs HEMTs by tri-layer T–shape gate'. In: *Microelectronic Engineering* 208 (2019), pp. 54–59.

[208] V. Bharath Sreenivasulu and Vadthiya Narendar. 'Design Insights of Nanosheet FET and CMOS Circuit Applications at 5-nm Technology Node'. In: *IEEE Transactions on Electron Devices* 69.8 (2022), pp. 4115–4122. DOI: 10.1109/ted.2022.3181575.

[209] Behzad Razavi. *Design of analog CMOS integrated circuits*. Mc Graw Hill, 2005.

[210] Benton H Calhoun et al. 'Digital circuit design challenges and opportunities in the era of nanoscale CMOS'. In: *Proceedings of the IEEE* 96.2 (2008), pp. 343–365.

[211] Gautam Biswas et al. 'Assessing design activity in complex CMOS circuit design'. In: *Cognitively diagnostic assessment*. Routledge, 2012, pp. 167–188.

[212] Pavan Kumar Kori et al. '22 nm LDD FinFET Based Novel Mixed Signal Application: Design and Investigation'. In: *Silicon* 14.15 (2022), pp. 9453–9465.

[213] Yaoyang Lyu et al. 'Machine Learning-Assisted Device Modeling With Process Variations for Advanced Technology'. en. In: *IEEE Journal of the Electron Devices Society* 11 (2023), pp. 303–310. ISSN: 2168-6734. DOI: 10.1109/jeds.2023.3277548.

[214] Bokyeom Kim and Mincheol Shin. 'A Novel Neural-Network Device Modeling Based on Physics-Informed Machine Learning'. en. In: *IEEE Transactions on Electron Devices* 70.11 (Nov. 2023), pp. 6021–6025. ISSN: 0018-9383, 1557-9646. DOI: 10.1109/ted.2023.3316635.

[215] Zeheng Wang et al. 'Improving Semiconductor Device Modeling for Electronic Design Automation by Machine Learning Techniques'. en. In: *IEEE Transactions on Electron Devices* 71.1 (Jan. 2024), pp. 263–271. ISSN: 0018-9383, 1557-9646. DOI: 10.1109/ted.2023.3307051.

[216] Gihun Choe, Jungyoun Kwak and Shimeng Yu. 'Machine Learning-Assisted Compact Modeling of W-Doped Indium Oxide Channel Transistor for Back-End-of-Line Applications'. en. In: *IEEE Transactions on Electron Devices* 71.1 (Jan. 2024), pp. 231–238. ISSN: 0018-9383, 1557-9646. DOI: 10.1109/ted.2023.3296715.

[217] Kumar Sheelvardhan et al. 'Machine Learning Augmented Compact Modeling for Simultaneous Improvement in Computational Speed and Accuracy'. en. In: *IEEE Transactions on Electron Devices* 71.1 (Jan. 2024), pp. 239–245. ISSN: 0018-9383, 1557-9646. DOI: 10.1109/ted.2023.3251296.

[218] Ya-Shu Yang, Yiming Li and Sekhar Reddy Reddy Kola. 'A Physical-Based Artificial Neural Networks Compact Modeling Framework for Emerging FETs'. en. In: *IEEE Transactions on Electron Devices* 71.1 (Jan. 2024), pp. 223–230. ISSN: 0018-9383, 1557-9646. DOI: 10.1109/ted.2023.3269410.

[219] Harsaroop Dhillon et al. 'TCAD-Augmented Machine Learning With and Without Domain Expertise'. en. In: *IEEE Transactions on Electron Devices* 68.11 (Nov. 2021), pp. 5498–5503. ISSN: 0018-9383, 1557-9646. DOI: 10.1109/ted.2021.3073378.

[220] Preslav Aleksandrov et al. 'Convolutional Machine Learning Method for Accelerating Nonequilibrium Green's Function Simulations in Nanosheet Transistor'. en. In: *IEEE Transactions on Electron Devices* 70.10 (Oct. 2023), pp. 5448–5453. ISSN: 0018-9383, 1557-9646. DOI: 10.1109/ted.2023.3306319.

[221] Hamilton Carrillo-Nuñez et al. 'Machine learning approach for predicting the effect of statistical variability in Si junctionless nanowire transistors'. In: *IEEE Electron Device Letters* 40.9 (2019), pp. 1366–1369.

[222] Kyul Ko et al. 'Prediction of Process Variation Effect for Ultrascaled GAA Vertical FET Devices Using a Machine Learning Approach'. en. In: *IEEE Transactions on Electron Devices* 66.10 (Oct. 2019), pp. 4474–4477. ISSN: 0018-9383, 1557-9646. DOI: 10.1109/ted.2019.2937786.

[223] Kyul Ko, Jang Kyu Lee and Hyungcheol Shin. 'Variability-Aware Machine Learning Strategy for 3-D NAND Flash Memories'. en. In: *IEEE Transactions on Electron Devices* 67.4 (Apr. 2020), pp. 1575–1580. ISSN: 0018-9383, 1557-9646. DOI: 10.1109/ted.2020.2971784.

[224] Jang Kyu Lee, Kyul Ko and Hyungcheol Shin. 'Prediction of Random Grain Boundary Variation Effect of 3-D NAND Flash Memory Using a Machine Learning Approach'. en. In: *IEEE Transactions on Electron Devices* 69.1 (Jan. 2022), pp. 447–449. ISSN: 0018-9383, 1557-9646. DOI: 10.1109/ted.2021.3130858.

[225] Gihun Choe et al. 'Machine Learning-Assisted Statistical Variation Analysis of Ferroelectric Transistor: From Experimental Metrology to Adaptive Modeling'. en. In: *IEEE Transactions on Electron Devices* 70.4 (Apr. 2023), pp. 2015–2020. ISSN: 0018-9383, 1557-9646. DOI: 10.1109/ted.2023.3244764.

[226] YS Bankapalli and HY Wong. 'TCAD augmented machine learning for semiconductor device failure troubleshooting and reverse engineering'. In: *2019 International Conference on Simulation of Semiconductor Processes and Devices (SISPAD)*. Ieee. 2019, pp. 1–4.

[227] Tong Wu and Jing Guo. 'Multiobjective Design of 2-D-Material-Based Field-Effect Transistors With Machine Learning Methods'. en. In: *IEEE Transactions on Electron Devices* 68.11 (Nov. 2021), pp. 5476–5482. ISSN: 0018-9383, 1557-9646. DOI: 10.1109/ted.2021.3085701.

[228] Jing Wang et al. 'Design of Terahertz InP pHEMT Using Machine Learning Assisted Global Optimization Techniques'. en. In: *2021 16th European Microwave Integrated Circuits Conference (EuMIC)*. London, United Kingdom: Ieee, Apr. 2022, pp. 67–70. ISBN: 978-2-87487-064-4. DOI: 10.23919/EuMIC50153.2022.9784068.

[229] Haoqing Xu et al. 'A Machine Learning Approach for Optimization of Channel Geometry and Source/Drain Doping Profile of Stacked Nanosheet Transistors'. en. In: *IEEE Transactions on Electron Devices* 69.7 (July 2022), pp. 3568–3574. ISSN: 0018-9383, 1557-9646. DOI: 10.1109/ted.2022.3175708.

[230] Chin-Cheng Chiang et al. 'Design and Process Co-Optimization of 2-D Monolayer Transistors via Machine Learning'. en. In: *IEEE Transactions on Electron Devices* 70.11 (Nov. 2023), pp. 5991–5996. ISSN: 0018-9383, 1557-9646. DOI: 10.1109/ted.2023.3310942.

[231] Sujit Kumar Singh et al. 'Quantum-Dot-Based Thermometry Using 12-nm FinFET and Machine Learning Models'. In: *IEEE Transactions on Electron Devices* (2024).

[232] M. Ehteshamuddin et al. 'Machine Learning-Assisted Multiobjective Optimization of Advanced Node Gate-All-Around Transistor for Logic and RF Applications'. en. In: *IEEE Transactions on Electron Devices* 71.2 (Feb. 2024), pp. 976–982. ISSN: 0018-9383, 1557-9646. DOI: 10.1109/ted.2023.3345288.

[233] Changwook Jeong et al. 'Bridging TCAD and AI: Its Application to Semiconductor Design'. In: *IEEE Transactions on Electron Devices* 68.11 (2021), pp. 5364–5371. DOI: 10.1109/ted.2021.3093844.

[234] Gennady Gildenblat, ed. *Compact Modeling*. Dordrecht: Springer Netherlands, 2010. ISBN: 978-90-481-8613-6. DOI: 10.1007/978-90-481-8614-3.

[235] TCAD Sentaurus. 'Sdevice user guide'. In: *V-2023.09, Synopsys* (2023).

[236] S. C. Rustagi et al. 'CMOS Inverter Based on Gate-All-Around Silicon-Nanowire MOSFETs Fabricated Using Top-Down Approach'. In: *IEEE Electron Device Letters* 28.11 (2007), pp. 1021–1024. DOI: 10.1109/led.2007.906622.

[237] Ahmet F. Budak et al. 'DNN-Opt: An RL Inspired Optimization for Analog Circuit Sizing using Deep Neural Networks'. In: *2021 58th ACM/IEEE Design Automation Conference (DAC)*. Issn: 0738-100x. Dec. 2021, pp. 1219–1224. DOI: 10.1109/dac18074.2021.9586139.

[238] Timothy P. Lillicrap et al. *Continuous control with deep reinforcement learning*. arXiv:1509.02971 [cs, stat]. July 2019.

[239] Hartmut Pohlheim. 'Examples of objective functions'. In: *Retrieved* 4.10 (2007), p. 2012.

[240] Ernesto P Adorio and U Diliman. 'Mvf-multivariate test functions library in c for unconstrained global optimization'. In: *Quezon City, Metro Manila, Philippines* 44 (2005).

[241] Ankit Dixit and Dip Prakash Samajdar. 'Extraction of performance parameters of nanoscale SOI LDD-FinFET using a semi-analytical model of capacitance and channel potential for low-power applications'. In: *Applied Physics A* 126.10 (2020), p. 782.

[242] Syed Samsuz Zaman et al. 'Design and simulation of SF-FinFET and SD-FinFET and their performance in analog, RF and digital applications'. In: *2017 IEEE International Symposium on Nanoelectronic and Information Systems (iNIS)*. Ieee. 2017, pp. 200–205.

[243] David Silver et al. 'Mastering the game of go without human knowledge'. In: *nature* 550.7676 (2017), pp. 354–359.

[244] Volodymyr Mnih et al. 'Playing atari with deep reinforcement learning'. In: *arXiv preprint arXiv:1312.5602* (2013).

[245] Ahmad EL Sallab et al. 'Deep reinforcement learning framework for autonomous driving'. In: *arXiv preprint arXiv:1704.02532* (2017).

[246] Volodymyr Mnih et al. 'Human-level control through deep reinforcement learning'. In: *nature* 518.7540 (2015), pp. 529–533.

[247] Changjun Fan et al. 'Searching for spin glass ground states through deep reinforcement learning'. en. In: *Nature Communications* 14.1 (Feb. 2023). Number: 1 Publisher: Nature Publishing Group, p. 725. ISSN: 2041-1723. DOI: 10.1038/s41467-023-36363-w.

[248] Kyle Mills, Pooya Ronagh and Isaac Tamblyn. 'Finding the ground state of spin Hamiltonians with reinforcement learning'. en. In: *Nature Machine Intelligence* 2.9 (Sept. 2020). Number: 9 Publisher: Nature Publishing Group, pp. 509–517. ISSN: 2522-5839. DOI: 10.1038/s42256-020-0226-x.

[249] Changjun Fan et al. 'Finding key players in complex networks through deep reinforcement learning'. en. In: *Nature Machine Intelligence* 2.6 (June 2020). Number: 6 Publisher: Nature Publishing Group, pp. 317–324. ISSN: 2522-5839. DOI: 10.1038/s42256-020-0177-2.

[250] Hanjun Dai et al. *Learning Combinatorial Optimization Algorithms over Graphs*. arXiv:1704.01665 [cs, stat]. Feb. 2018.

[251] Yutian Chen et al. 'Learning to Learn without Gradient Descent by Gradient Descent'. In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. Pmlr, Aug. 2017, pp. 748–756.

[252] Timothy Hospedales et al. 'Meta-Learning in Neural Networks: A Survey'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.9 (Sept. 2022). Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 5149–5169. ISSN: 1939-3539. DOI: 10.1109/tpami.2021.3079209.

[253] Nina Mazyavkina et al. *Reinforcement Learning for Combinatorial Optimization: A Survey*. arXiv:2003.03600 [cs, math, stat]. Dec. 2020.