T.R.
GEBZE TECHNICAL UNIVERSITY
GRADUATE SCHOOL

ARTIFICIAL NEURAL NETWORK-POWERED ANALOG/RF CIRCUIT DESIGN AUTOMATION

ENES SAĞLICAN

A THESIS OF MASTER OF SCIENCE
DEPARTMENT OF ELECTRONICS ENGINEERING
ELECTRONICS ENGINEERING PROGRAM

ADVISOR: ASSOC. PROF. ENGİN AFACAN

DECEMBER 2023

T.R.
GEBZE TECHNICAL UNIVERSITY
GRADUATE SCHOOL

ARTIFICIAL NEURAL NETWORK-POWERED ANALOG/RF CIRCUIT DESIGN AUTOMATION

ENES SAĞLICAN

A THESIS OF MASTER OF SCIENCE
DEPARTMENT OF ELECTRONICS ENGINEERING
ELECTRONICS ENGINEERING PROGRAM

ADVISOR: ASSOC. PROF. ENGİN AFACAN

DECEMBER 2023

T.C.
GEBZE TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

YAPAY SİNİR AĞI DESTEKLİ ANALOG/RF DEVRE TASARIM OTOMASYONU

ENES SAĞLICAN

YÜKSEK LİSANS TEZİ
ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI
ELEKTRONİK MÜHENDİSLİĞİ PROGRAMI

DANIŞMAN: DOÇ. DR. ENGİN AFACAN

ARALIK 2023

M.Sc. JURY APPROVAL FORM

A Thesis submitted by **Enes Sağlıcan** whose thesis defense exam was made on 13/12/2023 by the jury formed with the 11/12/2023 date and 2023/23 numbered decision of the GTU Graduate Administrative Board, was accepted as a **MASTER of SCIENCE** thesis in the **Department of Electronics Engineering, Electronics Engineering Program**.

JURY

JURY MEMBERS
(THESIS ADVISOR) : Assoc. Prof. Engin Afacan

MEMBER : Prof. Dr. Günhan Dündar

MEMBER : Assist. Prof. İhsan Çiçek

APPROVAL

Gebze Technical University Graduate Administrative Board
…..…/…..…/..……. date and ………/……... numbered decision.

SIGNATURE/MÜHÜR

ABSTRACT

The increasing demand for consumer electronic products pushes vendors into the massive production of integrated circuits. This aggressive demand for electronic products has necessitated the rapid launch of these products into the market. Electronic design automation approaches become inevitable under these circumstances. While the automation of analog/RF integrated circuits has been studied extensively over the years, the design optimization problem for industrial-standard circuits has not been fully resolved. With elevated technology and the speed of the modern world, our conventional ways to automate the design process of analog/RF circuits should catch up to the new trajectory. Considering this necessity we designers should not ignore and overlook the paramount importance of design accuracy in analog/RF circuits. Regarding these necessities, this thesis aims to show that ANN-powered multi/many-objective optimization tool increases the efficiency of the synthesis process. In this thesis, a simulation-based analog/RF integrated circuit optimizer tool has been initially conceived. To determine the metaheuristic algorithm to be used in the optimization kernel, two well-known evolution-based algorithms have been comprehensively compared with each other. Seven different basic analog/RF building blocks have been synthesized with the algorithm that proved its effectiveness. Upon identifying the most suitable algorithm for analog/RF circuit synthesis, the tool has been modified to incorporate sub-circuit modeling via artificial neural network modeling. Each low-level circuit has been trained in its optimization process and has been included in the synthesis. This modification not only expedited the optimization process but also paved the way for hierarchical optimization, leveraging the circuit models created. Given that a majority of analog/RF integrated circuits are inherently designed hierarchically to execute complex functions, this approach addresses a longstanding challenge in analog/RF automation. In this context, an analytical model-based hierarchical synthesis method has been also introduced in the thesis. The multistage amplifier circuit has been synthesized in a hierarchical structure within seconds with the help of the presented method. Additionally, the effect of the preamplifier circuit on the performance of the voltage comparator has been examined with a similar method.

Keywords: Electronic Design Automation, Optimization, Analog/RF Integrated Circuits, Evolutionary Algorithm, Artificial Neural Networks, Hierarchical Design

ÖZET

Son kullanıcı elektronik ürünlerine yönelik artan talep, satıcıları tümdevreler için kitlesel üretime yönlendirmektedir. Söz konusu bu yoğun talep, ürünlerin piyasaya hızla sunulmasını zorunlu kılmaktadır. Tüm bu koşullar göz önünde bulundurulduğunda elektronik tasarım otomasyonu yaklaşımları hayati bir önem taşımaktadır. Analog/RF entegre devrelerin otomasyonu yıllardır yoğun bir şekilde çalışılmış olsa da endüstri standardı devreler için tasarım optimizasyon sorunu tam olarak çözülememiştir. Yükselen teknoloji ve modern dünyanın hızıyla birlikte, geleneksel tasarım yöntemlerimizin günümüz isterlerine hızlı bir şekilde cevap vermesi için otomasyona ağırlık verilmelidir. Bu zorunluluğu düşünerek, tasarımcılar olarak analog/RF devrelerde tasarım doğruluğunun üstün önemini ihmal etmemeliyiz. Bu gereklilikler göz önüne alındığında, tez, yapay sinir ağı destekli bir optimizasyon aracının sentez sürecini hızlandırabileceğini göstermeyi amaçlamaktadır.Tezde ilk olarak benzetim tabanlı bir analog/RF tümdevre otomasyon aracı tasarlanmıştır. Eniyileme aracının çekirdeğini oluşturacak metasezgisel algoritmayı belirlemek için iyi bilinen iki evrim tabanlı algoritma birbirleriyle kapsamlı bir şekilde karşılaştırılmıştır. Yedi farklı temel analog/RF alt devre, etkinliği kanıtlanmış algoritma tarafından sentezlenmiştir. Analog/RF devre sentezi için en uygun algoritmanın belirlenmesinden sonra araç, yapay sinir ağları kullanılarak alt seviye devrelerin modellenmesini de içerecek şekilde güncellenmiştir. Her bir alt seviye devre, kendi eniyilenmesi sürecinde eğitilerek senteze dahil edilmiştir. Bu değişiklik, sadece eniyileme sürecini hızlandırmakla kalmamış, aynı zamanda oluşturulan devre modellerini kullanmak suretiyle hiyerarşik eniyilemenin de yolunu açmıştır. Analog/RF tümdevrelerin çoğunlukla karmaşık işlevleri yerine getirebilmesi için hiyerarşik olarak tasarlandığı düşünüldüğünde, bu yaklaşım analog/RF otomasyonunda uzun süredir devam eden bir sorunu ele almaktadır. Bu bağlamda, tezde analitik model-tabanlı bir hiyerarşik sentezleme yöntemi de tanıtılmaktadır. Çok katlı yükselteç devresi sunulan yöntemle saniyeler içerisinde hiyerarşik yapıda sentezlenmiştir. Ayrıca, ön-yükselteç devresinin gerilim karşılaştırıcının performansı üzerindeki etkisi yine benzer bir yöntem ile incelenmiştir.

Anahtar Kelimeler: Elektronik Tasarım Otomasyonu, Eniyileme, Analog/RF Tümdevreler, Evrimsel Algoritma, Yapay Sinir Ağları, Hiyerarşik Tasarım

ACKNOWLEDGEMENTS

Foremost, I wish to extend my sincere gratitude to my esteemed advisor, Assoc. Prof. Engin AFACAN, whose generosity in sharing profound technical insights and unwavering support has been an invaluable source of guidance and encouragement throughout my academic journey. Additionally, I express my profound appreciation to Prof. Dr. Günhan Dündar for his steadfast support, which has greatly contributed to the success of my endeavors. Also, I express my gratitude to Assist. Prof. İhsan Çiçek for his invaluable contributions as a member of the thesis evaluation committee. In moments of perplexity where I found myself grappling with certain intricacies, his unconventional ideas have consistently broadened my vision.

Of course, my deepest gratitude goes to my family. I would like to express my endless gratitude to my parents Emel and Namık, my brother Emre, and his wife Büşra, who is my sister-in-law, for their support and love. It was truly impossible for me to achieve some things without their support.

I would also like to thank my friends Ubeyd, Abdullah, and Hakan, at our laboratory, LAMBDA, for their continuous support.

This study was funded by TUBITAK ARDEB 3501 under the project number 121E430.

TABLE OF CONTENTS

Page

LIST OF SYMBOLS AND ABBREVIATIONS

ADC	:	Analog-to-Digital Converter
ANN	:	Artificial Neural Network
BGR	:	Bandgap Reference
CAD:	:	Computer-Aided Design
CMOS	:	Complementary Metal-Oxide Semiconductor
DHR	:	Derived Hyperarea Ratio
DL	:	Deep Learning
DNN	:	Deep Neural Network
EA	:	Evolutionary Algorithm
EC	:	Evolutionary Computation
EDA	:	Electronic Design Automation
EM	:	Electromagnetic
ES	:	Evolutionary Strategies
GA	:	Genetic Algorithm
GD	:	Generational Distance
HR	:	Hyperarea Ratio
HV	:	Hypervolume
IC	:	Integrated Circuit
IFA	:	Intermadiate Frequency Amplifier
IGD	:	Inverted Generation Distance
LNA	:	Low-Noise Amplifier
ML	:	Machine Learning
MOEA/D	:	Multi-Objective Evolutionary Based on Decomposition
MOO	:	Multi-Objective Optimization
MOSFET	:	Metal Oxide Semiconductor Field Effect Transistor
MS	:	Maximum Spread
NSGA-II	:	Non-dominated Sorting Genetic Algorithm
OPAMP	:	Operational Amplifier
OTA	:	Operational Transconductance Amplifier
PM	:	Phase Margin
PoF	:	Pareto-optimal-Front
PSO	:	Particle Swarm Optimization
PWL	:	Piecewise Linear
RL	:	Reinforcement Learning
SA	:	Simulated Annealing
SI	:	Swarm Intelligence
SOO	:	Single Objective Optimization
SP	:	Spacing
SPEA-2	:	Strength Pareto Genetic Algorithm - 2
SSFDA	:	Single Stage Fully Differential Amplifier
SVM	:	Support Vector Machines
SVR	:	Support Vector Regression

LIST OF FIGURES

LIST OF TABLES

Page

1. INTRODUCTION

The utilization of electronic design automation (EDA) tools for integrated circuit (IC) synthesis has gained more popularity over the years. Before the development of these tools, the circuit had been designed and laid out manually by a designer. Firstly design process was automated, and following it, layout considerations were handled by computers at the end of the 1970s. Enhancing computer-aided design (CAD) with resourceful programming languages can be considered a milestone for EDA approaches for VLSI systems in the early 80s. The design time is saved as a result of the increasing accessibility of EDA tools; on the other hand, the complexity of ICs has increased exponentially due to the need for circuits with more functionality. Over the years, CAD tools have played more crucial role from the design phase to manufacturing. As a result of aggressive demand for consumer electronics, especially in the last years, the chip manufacturing industry has tended to increase production volume. Combining of currently growing production volume and hard challenging requirements for nowadays ICs, design automation is an inevitable solution to minimize human effort and time consumption.

Design automation is a concept that solves electrical/physical-level circuit design problems through highly intelligent algorithms without the need for human effort. Although the automation of digital circuits has been well-established, analog/RF circuits have not been fully automated due to their highly non-linear nature. For instance, during the digital circuit synthesis, the speed is maximized, while chip area and power are minimized. In contrast, analog integrated circuits have tough design requirements involving tremendous trade-offs. In this context, analog/RF IC automation process is still developing and it contains a lot of gray areas that must be enlightened.

Design automation of analog circuits has been in progress for a few decades. The first developed automation tools needed human expertise to synthesize proper analog circuits. As the analog/RF circuits are highly non-linear, designer dependency was acceptable for these tools. Over the years, with the increasing of computational capability of computers and pushing the limits of intelligence algorithms, new horizons have been seen for analog/RF circuit synthesis. Human dependency was reduced and efficiency was increased, and this progress has encouraged researchers to solve automation problems of more complex analog/RF circuits. As a result, a new chapter has begun with employing metaheuristic algorithms in analog/RF design automation. Metaheuristic algorithms have become broadly used in the circuit synthesis process due to their efficiency, reliability, and robustness over the years. Many synthesis tools that utilize meta-heuristic algorithms have been developed to overcome analog/RF design problems, from schematic sizing to reliability issues for manufacturing. These algorithms have taken a place in solving of quietly different analog/RF circuit design problems as shown in **Figure 1.1**.

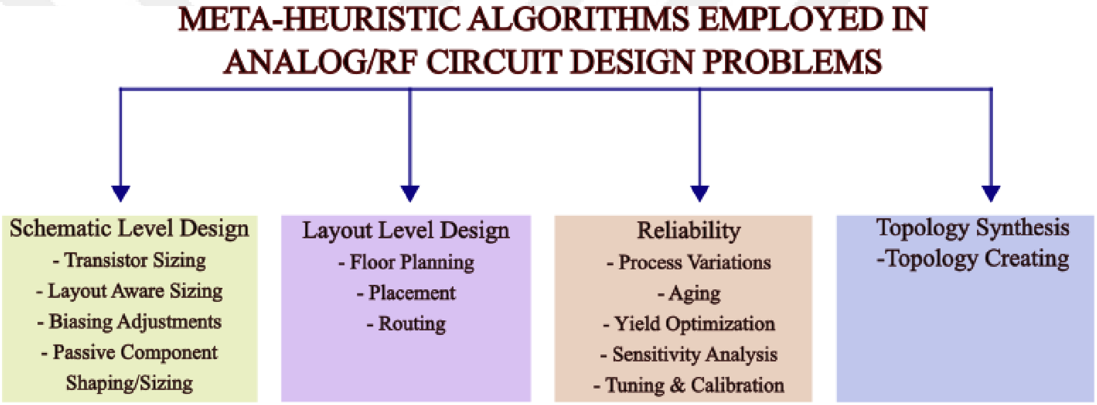


Figure 1.1: Metaheuristic algorithms in analog/RF circuit automation problems.

After providing problem-solving ability of meta-heuristic algorithms in analog/RF circuit optimization, the researchers have sought new considerations about circuit synthesis. Analog/RF circuit synthesis tools have suffered from two different troubles which are not considered at the time: Accuracy and synthesis time. Basically, these intelligent algorithms start with randomly generated population, and they generate new solutions using the information of the previous solution set. For evaluation, it needs a performance evaluator to measure the circuit performance for specific design parameters. Conventionally, the synthesis has been carried out by equation-based or simulation-based approaches. While time is saved by utilizing a set of equations to estimate the performance of the circuit, the accuracy of the synthesis results is still arguable. Also, analytical equations are not adequate to mimic more complex circuit behavior. Therefore, most analog/RF automation tools have utilized circuit simulators to evaluate circuit performance. In summary, the accuracy problem is preferred for the sake of efficiency. However, the synthesis time of complex circuits including several sub-circuits in the hierarchy, has exceeded beyond the practical limits. In this context, a question arises: Is it possible to reduce the synthesis time without sacrificing accuracy? Thanks to nowadays' powerful computers and advances in machine learning, a door has been opened to answer this question. Since different machine learning techniques have been successfully applied to solve many different engineering or real-world problems, researchers have found the strength to try these techniques on analog/RF circuit design problems. Machine learning techniques such as support vector machines (SVM), artificial neural network (ANN), polynomial regression, deep learning (DL), multi-layer perceptron (MLP), and reinforcement learning (RL) have been successfully utilized in analog/RF circuit synthesis to fulfill design, layout, and test considerations. However, in the literature, machine learning algorithms have been only applied to solve the design problem of a few basic analog/RF sub-circuits such as amplifier, current conveyor, or voltage comparator. It seems that the ML techniques in analog/RF circuit synthesis need to be optimized; thus, ML techniques-based synthesis would become a popular research topic in this field.

This thesis provides an analog design automation tool that has been strengthened with ANN assistance. Highly accurate models of sub-block circuits have been generated during the synthesis process and these models have been employed in hierarchical design. Therefore, hierarchical synthesis would be performed efficiently and the creation of the ANN model for a whole system would be possible within a short time. The thesis is organized as follows. In Section 2, a detailed literature review on analog/RF circuit automation is provided. In Section 3, a background is introduced for analog/RF circuit design automation. In Section 4, EA algorithms' performances are discussed. In Section 5, ANN-based model generation and circuit optimization are provided. In Section 6, hierarchical design synthesis approaches for analog/RF systems are introduced. Finally, Section 7 concludes this thesis by providing general remarks and future work.

2. PREVIOUS WORK

The idea of automation of analog/RF integrated circuits dates back to the few decades. The first developed tools mostly relied on human expertise, which are more suitable for basic sub-blocks. In [1], an equation-based optimization tool have been introduced for operational amplifier synthesis. Amplifier topology have been divided into several parts: The current mirror, the input pair, and the load. Another equation-based tool have been introduced in [2] that creates the circuit topology. The predicted performance of synthesized topology has been fixed, if necessary, with the aid of current correction method, and synthesized circuit verified with electrical simulation. A design tool called IDAC, introduced in [3] has the ability of the layout file generation. The authors have claimed that IDAC could be considered as an assistant for an analog designer. OPASYN [4] provides topology selection for amplifiers. A proper topology has been selected from the topology library presented in tool. The circuit performances have been evaluated by analytical equations, therefore the high need of designer knowledge has aroused as increasing circuit complexity. Another CMOS-OPAMP compiler GPCAD [5] uses genetic programming for six CMOS OPAMP circuit design. Since the tool is equation based, the proper topology has been selected within just a few seconds for a given design specifications. The tool can also work properly with different technology parameters.

With the first improvements in the analog circuit automation, researchers have focused on the way of finding the optimal solutions. The main consideration was the location of the obtained solutions in the whole possible design space for a given parameter boundaries. As a solution, OPTIMAN in [6] has used simulated annealing (SA)[7] to get rid of local optimum at the expense of computation time. To reduce the total optimization time, the tool have leveraged a symbolic simulator. An OPAMP synthesis tool, named DARWIN [8], is used genetic algorithm as optimization kernel. Both topology selection and circuit sizing have been concurrently carried out with genetic algorithm. The performance of the circuit, where the topology is selected from 24 available OPAMP topologies stored in the tool, has been evaluated by the small-signal equivalent circuits and analytical expressions. The initial solutions have lightly constrained, and as generations progressed, the design constraints to achieve global optimal solutions become more difficult. The optimization process have completed when the maximum generation count was reached. Also, the tool explained in [9], a folded cascode OTA and an analog filter have been optimized with the genetic algorithm, where analytical expressions were utilized as performance estimator.

There is compelling a trade-off for the analog/RF design automation process: accuracy and efficiency. The first simulator-based optimization tool DELIGHT.SPICE [10] has improved optimization efficiency. On the other hand in, FASY [11], proper topology for the determined design specifications has been selected with fuzzy-logic method [12]. Decision rules used in the topology selection have been introduced by an expert designer or automatically generated via learning process with back-propagation method. The circuits have synthesized via two consecutive optimizers. The first one searching for global optimum is simulated annealing-based optimization where equations considered as the circuit evaluator. While time saving was provided by the first optimizer, the second one is simulation-based to ensure the accuracy. ANACONDA [13] utilized stochastic pattern search where commercial simulators was run in parallel for the time efficiency. Also, process variations were taken into account with a few light constraints. In [14], the combination of evolutionary algorithm and simulated annealing has been introduced as the optimization engine. SA has been utilized for avoiding of the local optimum solution. On the other hand, evolution strategies (ES) [15] have searched for the feasible solutions. The circuit performances have been evaluated by a fast DC simulator and user-defined equations/neural fuzzy performance models for DC parameters and AC parameters, respectively. Automation process has been validated with silicon measurements for two amplifiers and a current conveyor circuit. Again, for robust global search in the solution space, a genetic/annealing algorithm have presented in [16]. Parallel simulations have been conducted on different workstations and simulator details have been hidden by a software layer for the time efficiency. Three different OPAMP circuits have been sized by the tool.

Increasing computing capacity of computers has led researchers to develop fully simulation - based optimizers. In GENOM [17], a multi-objective and multi-constrained analog optimization problem solver with genetic algorithm has been presented for a folded-cascode amplifier. To avoid the local optima, the mutation rate has been increased if there is no remarkable progressing in consecutive generations. Another SPICE-based optimizer [18] presents an optimum amplifier topology from topology library included by the user as well as circuit sizing. To avoid premature convergence age level population structure (ALPS) [19] has injected to the optimization kernel where Non-Dominated Sorting Genetic Algorithm - II (NSGA-II) [20] is the search algorithm. The whole design space is explored by the search algorithm has been provided both multi and many-objective optimization problem for CMOS OPAMPs. In [21], four different OPAMP circuit have been automated via an evolution-based methodology. To achieve the optimal single-objective solution with satisfying desired design constraints, a memetic single-objective evolutionary algorithm (MSOEA) has been utilized as the searching algorithm.

It was seen that automation of analog ICs have become more efficient with the SPICE integration and use of analytical equations. Nevertheless, the first attempts are mostly on amplifier optimization or circuits having less complexity. In these cases, researchers have focused on effective and reliable optimization methods to synthesize complex analog/RF circuits. It is worth note that, as always all studies have focused on to find the best point with establishing a balance between the accuracy and the efficiency. In [22], a multi-objective optimization tool has been proposed for performance space exploration through obtained pareto optimal front (PoF). The simulated annealing-based multi-objective optimization algorithm has been employed to find optimal solution set. An active filter and two amplifiers have been synthesized where layout-aware circuit matrix modeling is performance estimator. Another multi-objective simulation-based optimization tool GENOM-POF [23] sizes two amplifier circuits considering process variations. Since the circuit netlist is provided to the optimizer by the user, the synthesis tool is independent of the circuit or the topology. On the other hand, the authors have pointed out that NSGA-II is quite good at creating PoF compared to other multi-objective evolutionary algorithms. Swarm intelligence (SI) algorithms are also used in analog/RF circuit optimization problems. Equation-based optimization was provided in [24] where particle swarm optimization [25] was introduced as an optimization engine to solve mono and multi-objective sizing problems. An RF circuit and an analog circuit have been chosen as the case studies. Another tool leveraged by particle swarm optimization technique given in [26] sizes a differential amplifier and two-stage amplifier through a single-objective optimization approach. The solutions obtained from the synthesis process have been validated by SPICE simulations. In [27], ant colony optimization [28] which mimics foraging patterns of some ant species is an optimization kernel. The optimization process evolves according to the simulation results generated in each iteration. To validate the reliability of ant colony optimization, the authors also have compared it with the genetic algorithm. An analog voltage buffer, a CMOS Miller OTA, and a two-stage OPAMP have been synthesized. As the circuit complexity increased, design constraints were also stringent. In addition, sub-blocks of an analog-to-digital converter have been optimized and ADC has been constructed by combining these synthesized blocks. In [29], the new algorithm called CODE is presented for analog IC synthesis. The proposed algorithm has been compared with two other evolutionary algorithms via synthesis results of three different amplifiers. The circuits whose topology has been determined by the tool also have been automatically sized. The algorithm has been implemented in MATLAB, and the circuit behavior has been estimated via HSPICE. The simulation-based optimizer presented in [30] leverages simulated annealing and particle swarm optimization combination to avoid local optima and forming the PoF, respectively. Also, many-objective optimization has been performed to obtain a parato surface for Miller amplifier. In [31], g_m/I_d technique with designer knowledge helps to speed up the optimization process without sacrificing accuracy. All synthesized circuits were validated by SPICE simulations. If a significant error occurs, device parameters update themselves according to the simulation result. Folded cascode OTA and two-stage OTA circuits have been sized when supply voltage was 0.6V, 1.2V, and 1.8V to ensure operating in weak, moderate, and strong inversion regions, respectively. The equation-based analog/RF circuit automation method proposed in [32] includes quantitatively PoF quality comparison obtained via different swarm intelligence algorithms and NSGA-II. Those interested can find several PoFs for the current conveyor, CMOS OPAMP, and LNA. In [33], an LC voltage-controlled oscillator has been synthesized with equation-based optimization where GA searches solution space. Circuits have been sized via a single-objective optimization approach. On the other hand, ABSYNTH [34], a multi-featured optimization tool that has an interface solves multi-objective sizing problems for analog circuits. The optimization process has been separated into three steps where circuit performance evaluation is carried out by a support vector regression (SVR) model, NGSPICE, and a commercial simulator OCeaN. Particle swarm optimization or harmony search (HS) has been utilized in the parts of the whole automation process. The time is saved utilizing the self-learning method also avoids redundant data. Also, the authors have indicated that more efficient results can be obtained via a combination of algorithms compared to the flat simulation-based particle swarm optimization. According to the authors' best knowledge, a butterfly-inspired optimization algorithm has been proposed for analog circuit synthesis for the first time in [35]. Equation-based optimization has been carried out for both single- and many-objective sizing problems for analog amplifiers. In [36], the developed tool generates a proper topology and designs the circuit. The functionality of the generated topology has been checked via topological symbolic analysis based on g_m/I_d methodology and curve fitting technique. If the topology passes the first checking procedure, the devices are sized otherwise, the topology generation procedure runs again. The efficiency of the synthesis has been enhanced through the implementation of a checking procedure before sizing. [37] presents a modified NSGA-II for analog/RF circuit automation. The algorithm that differs from the original by the inclusion of a hierarchical mutation procedure has been compared with the other NSGA-II versions both with benchmark problems and real analog/RF IC sizing problems. While the modified algorithm performs head-to-head at benchmark problems, on the other hand, it outperforms other NSGA-II versions in terms of design space exploration for circuit sizing problems. Similar to the previous one, in [38] the better quality PoFs have been obtained with some improvements on the main algorithm.

Development of robust and accurate design automation tools has encouraged researchers to find new solutions to efficiently synthesize more complex analog/RF circuits and systems. In [39], pareto optimal surface via many-objective optimization is created for inductor, where electromagnetic (EM) simulation is utilized as a performance estimator. In the sizing phase of the LNA circuit, the inductors are selected from the solution pool in the obtained pareto surface. Thus, quite accurate synthesis can be achieved without excessive optimization time beyond practical limits. Single- and many-objective synthesis examples have been introduced. Parasitic-aware sizing methodology is presented in [40]. While floorplan constraints, performance constraints, and layout parasitics are considered in the genetic programming (GP) at the sizing phase, synthesis accuracy is increased with the evolutionary algorithm (EA) which is communicated to electrical simulation. The equation-based optimization technique with GP not only increased the efficiency of the optimization but also facilitated the resolution of the problem in the EA-based second phase. The presented sizing methodology was validated via post-silicon measurements for the two-stage amplifier, voltage comparator, and LNA. Obtained design through parasitic-aware optimization was more immune to the parasitic mismatch which can drastically cause performance degradation. Another method that considers speeding up the optimization process and increasing efficiency can be found in [41]. The author has claimed that determining the operation region for a transistor with simple analytic region conditions is not adequate for real devices. Therefore, the inversion coefficient has been used to determine the actual operation region. For each case study (amplifier, LNA, and oscillator synthesis) the most efficient inversion coefficient intervals were determined and defined as a constraint in the optimization problem. Both fast convergence and better cost function have been achieved by dint of this method. In [42], different cross-coupled CMOS LC oscillator topologies are compared utilizing obtained PoF through multi-objective optimization. A simulation-based, layout-aware design automation tool for RF circuits has been developed using strength pareto evolutionary algorithm - 2 (SPEA-2) [43]. The ”2 - π” models have been utilized for the passive element for the sake of increasing accuracy. Synthesis results are validated silicon data. In [44], the mutation process is manipulated via a designer perspective instead of purely mathematical operations. Pretty complex two analog amplifiers and an oscillator circuit have been synthesized via a genetic algorithm, where the mutation process is modified. With the proposed method four-stage amplifier circuit has been synthesized successfully, however, the genetic algorithm, with the unmodified mutation operator, has failed to converge.

Artificial intelligence using machine learning techniques has proven its problem-solving ability in various engineering fields and has led researchers to explore new automation methods for analog/RF circuits automation. A yield-aware circuit synthesis tool [45] collaborates with a deep neural network (DNN) to optimize analog circuits while considering process variation. As the time spent during simulation-based Monte Carlo (MC) analysis is impractical for the synthesis process, deep neural network models have been trained and utilized in an optimization loop instead of SPICE. The authors have indicated that synthesis has accelerated by 29x, with the inclusion of DNN in the optimization process for the synthesis of the single-stage amplifier, the two-stage amplifier, and the voltage comparator. In [46], DNN models have been used to increase synthesis process speed for local minimum search, which aims to improve global optimization results. Different DNN models have been trained for all design specifications(gain, bandwidth, slew rate, etc.). Four different operational amplifiers are synthesized with strict constraints where the optimization process has sped up x2 with the help of DNN models. In [47], a Gaussian-process based surrogate models have been utilized to accelerate the optimization loop. Modeling of some device phenomenons like the short channel effect or narrow channel effect is difficult by curve fitting techniques or analytical equations. Therefore, the authors proposed Gaussian-process regression modeling through the characterization of the technology, which included information about both nominal and corners. An active filter, a voltage regulator, and an oscillator were synthesized via a surrogate-based tool where computational cost decreased drastically. A local surrogate-based parallel optimization is proposed in [48]. The authors emphasize that evolutionary algorithms based optimization is not efficient even though it is effective. With the developed method, the proposed optimization achieves better solutions in a shorter time when compared with other parallel two metaheuristic algorithms and a state-of-art surrogate model-based optimization. Another optimization methodology in [49] exploits g_m/I_d fitting curves and the proposed current density factor to create a circuit model. The optimization process was conducted in two phases. Parasitic effects were taken into account in the first phase, which is a combination of symbolic modeling and mixer integer non-linear programming. The second phase is the sizing phase, where the simulation-based optimizer utilizes an evolutionary algorithm as the searching algorithm. Recently, artificial neural network (ANN) [50] has been utilized for analog/RF circuit modeling. In [51], the authors emphasized the possible lack of analog/RF circuit modeling via trained neural networks and they offer a new ANN model construction method which is implemented in the proposed optimization tool named ESSAB. The developmed tool has been tested with the synthesis of highly complex circuits, where design specifications have been stringently constrained. Recent studies show that the Bayesian optimization method can be applicable to analog/RF circuit optimization. Developed tools in [52–54] prove that the Bayesian optimization method is proper for analog circuit automation.

3. BACKGROUND

Optimization is a term about searching for the best way to solve any defined problem. It plays an important role in various scientific areas such as engineering, data analytics, and computer science. Many real-world problems inherently involve many requirements with challenging trade-offs among them. The existing problem must be solved by establishing a good balance among all requirements through an optimization procedure. In general, the optimization process is categorized under three sub-sections: Single-objective optimization (SOO), multi-objective optimization (MOO), and many-objective optimization. While in SOO, only one function or a mathematical expression of two or more functions is determined as an objective that is tried to be maximized or minimized, the multi-objective optimization searches the best solutions between two conflicting objective functions. On the other hand, three or more functions are defined and minimized or maximized in many-objective optimization. As analog/RF circuits have been automated via the multi-objective optimization approach in the thesis, the multi-objective optimization is introduced in this section.

3.1. Multi-Objective Optimization

Multi-objective optimization is a concept that minimizes or maximizes two functions by the solution X along with defined constraints that must be satisfied. The multi-objective optimization can be represented as Equation 3.1, where X is vector decision variables bounded by the interval between x_i^L and x_i^U which introduce lower and upper boundaries, respectively. Objective space Z for the M objective functions $f_m(X)$ is formed by solutions obtained from decision variable space D which is determined by x_i^L and x_i^U, as illustrated in **Figure 3.1**. Many real-world problems contain equality constraints or inequality constraints denoted as $h_k(X)$ and $g_j(X)$, respectively. Solutions that meet the defined constraints constitute feasible objective space.

On the contrary of SOO, the superiority of the solutions is determined by dominance in an MOO problem instead of comparing function values. Let us have two solutions x_1 and x_2 in feasible design space. The solution x_1 dominates x_2 if the solution x_1 is better than x_2 in all objectives or at least one objective of x_1 is better than the solution x_2.

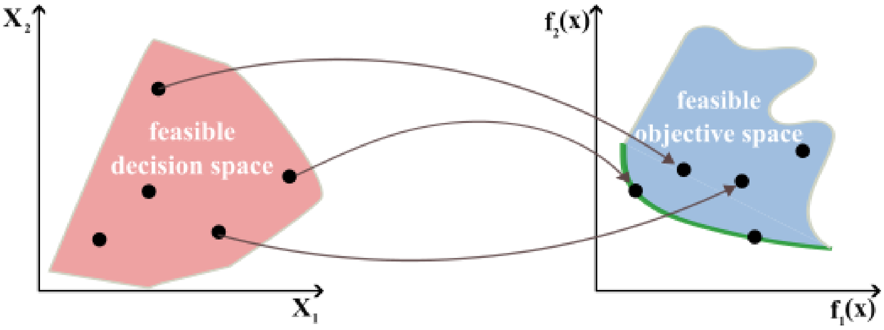


Figure 3.1: Mapping from decision space to objective space.

$$\begin{array}{rccc}
Minimize/Maximize & f_m(X), & m=1,2,3,...,M; \\
Subject\ to & g_j(X)\geq 0\ or\ g_j(X)\leq 0, & j=1,2,3,...,J; \\
 & h_k(X)=0, & k=1,2,3,...,K; \\
 & x_i^L\leq x_i\leq x_i^u, & i=1,2,3,...,N.
\end{array} \tag{3.1}$$

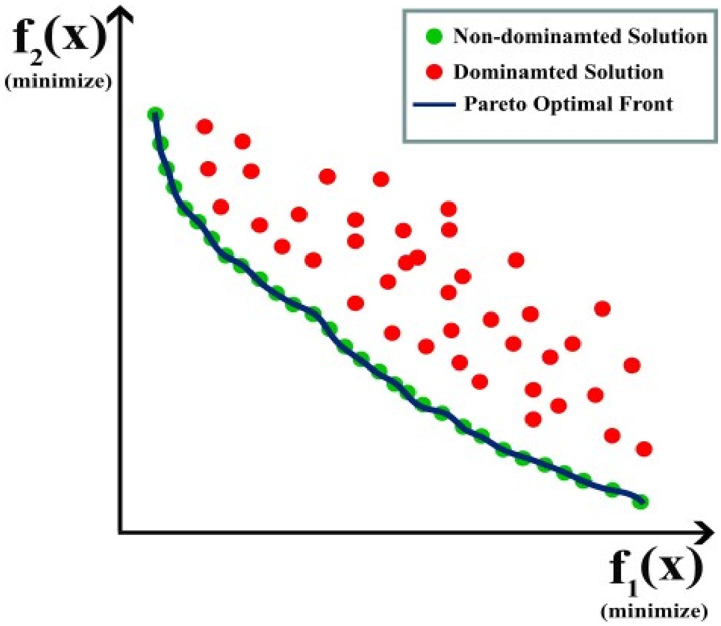


Figure 3.2: Graphical description of pareto optimal front.

Non-dominated solutions are not dominated by any solution in a given whole solution set. In the entire objective space, MOO promises to create the non-dominated solution set which is called PoF as illustrated in **Figure 3.2**. MOO aims to find a set of solutions as close as possible to the PoF without sacrificing diversity.

One of the best ways to realize multi-objective optimization is using metaheuristic algorithms that mimics swarm's communications between a group of insects or evolutionary theory. The genetic algorithms (GA) are inspired by natural selection and sexual production which are the essential evolving processes of organisms [55]. In the conventional GA, randomly generated individuals in the first population are mated to produce offspring via genetic operators such as mutation, cross-over, and recombination. The evolution process continues with iterations, in which individuals fight each other to survive. In each iteration, also called generation count, the produced offspring undergo a selection mechanism to select the best of them as the new parents. It is hoped that the new individuals in the next generation have better specifications compared to their parents.

3.2. Developed Tool for Multi-Objective Optimization of Analog/RF Circuits

Many tools have been developed in the literature to address the automation of the analog/RF circuit sizing problem. Metaheuristic algorithms are mostly preferred to solve IC sizing problems due to their robustness, even though they have high computational complexity. Thanks to powerful computers, the computational complexity of nature-inspired algorithms has not been the main concern. Moreover, researchers endeavors to reduce time with new methods that can replace the circuit simulator, which takes a large portion of the optimization time.

The general framework of the developed automation tool is depicted in **Figure 3.3**. Analog/RF circuits have been synthesized via multi-objective optimization tool in [56] that have been reconstructed to perform IC optimization. Non-dominated Sorting Genetic Algorithm-II (NSGA-II) was selected as the optimization kernel.

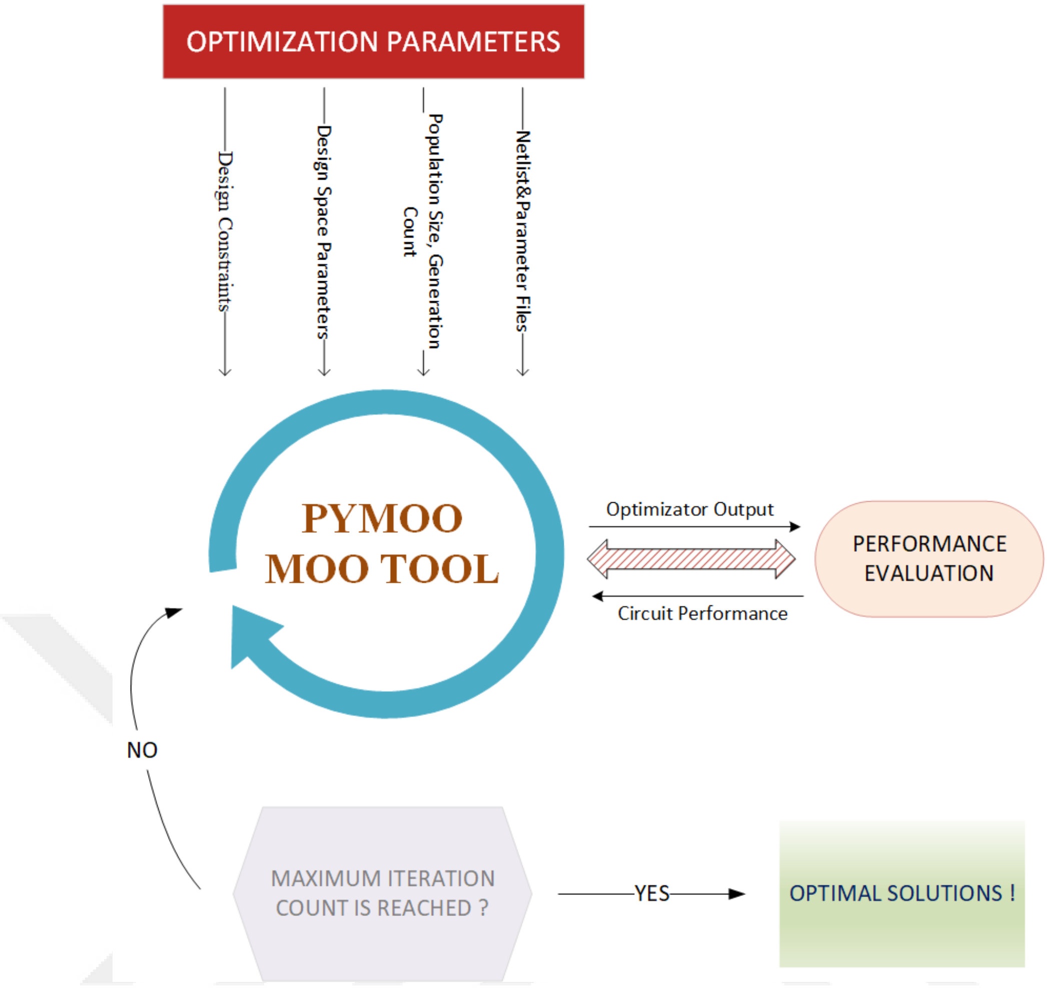


Figure 3.3: General framework of developed tool.

Before the optimization, the user determines design constraints, the boundaries of decision space, and the design objectives. Additionally, the netlist and design parameters file of the circuit must be provided beforehand. Objective normalizers, the number of populations, and the number of generations must be defined to initiate the optimization process. Optimization starts with the first generation where individuals are generated randomly. Each individual undergoes a performance evaluation process carried out by SPICE or a circuit model. In each cycle, the performance results are obtained as a result of the evaluation process. Crossover, mutation, and selection are applied to individuals to enter the optimization cycle and create a new generation. The circuit optimization process ends when the maximum number of iterations is reached.

3.3. Searching Algorithm: NSGA-II

NSGA-II is a modified approach of traditional GA, whose pseudo-code is given in **Table 3.1**. The algorithm begins with creating the first population. Let us say the population size in the first generation is N. Then, the population expands with created offspring where mutation, recombination, and selection operations are applied in the production phase; hence, the population size becomes 2N. All individuals in the new population (2N) are sorted according to their own non-domination rank. All non-dominated solution sets, where the best is F_1 and the worst is F_k are selected according to their ranks until the N individuals are obtained. Generally, the number of total solutions in all sets from F_1 to F_k is larger than the population size. The new population is filled with the ranked solution set until it reaches size N. If the population size exceeds with last non-dominated solution sets, to choose exactly N population members, the solutions of the last front F_l are sorted by crowding distance operator, and the best solutions are chosen as population members. The general procedure of NSGA-II is shown in **Figure 3.4**.

$$CD = \sum_{2}^{N-1} \frac{f_{i+1} - f_{i-1}}{f^{max} - f^{min}} \tag{3.2}$$

Crowding distance refers to the perimeter estimating of the formed cuboid around two neighboring solutions. The crowding distance for the objective in a pareto optimal set is calculated using Equation 3.2, where the first and the last solutions of the pareto optimal set are considered to be infinite. The maximum and minimum values of the objective space are denoted as f^{max} and f^{min}, respectively. f_i refers to the ith individual set of pareto-optimal solution.

$$\left\{ \begin{matrix} i \prec_n j \quad \text{if}\, i_{rank} < j_{rank} \\ or((i_{rank} = j_{rank}) \\ and(i_{distance} > j_{distance})) \end{matrix} \right\} \tag{3.3}$$

The crowding distance operator helps to ensure diversity and obtain uniformly distributed PoF by guiding the selection mechanisms at various stages of the algorithm.

According to Equation 3.3, the better solution (lower rank) is selected when two solutions do not belong to the same front. On the other hand, a solution located in a lesser crowded region is selected if both solutions are the members of the same front.

Table 3.1: The pseudo code of NSGA-II.

```
Input: parent $\mathcal{P}_t$, offspring $\mathcal{Q}_t$, number of generation $\mathcal{G}$, non-dominated
solution set $\mathcal{F}_i$
Output: survivors $\mathcal{S}$
Set initialize population randomly
Create offspring population $\mathcal{Q}_0$ and apply binary tournament selection,
recombination, and mutation
for $g = 0$ to $\mathcal{G}$ do:
    $\mathcal{R}_t = \mathcal{P}_t \cup \mathcal{Q}_t$
    Do non-dominated sorting
    $\mathcal{P}_{t+1} = \emptyset$ and i=1
    while $|\mathrm{P}_{t+1}|+|F_i| \leq \mathrm{N}$ do:
        Calculate the crowding distance of the front
        $|\mathrm{P}_{t+1}|=|P_{t+1}|\cup \mathcal{F}_i$
        i=i+1
    end while
    Do crowding distance sorting
    $|\mathrm{P}_{t+1}|=|P_{t+1}|\cup \mathcal{F}_i[1:(N-|\mathrm{P}_{t+1}|)]$
    Use selection, crossover and mutation to create new population $\mathcal{Q}_{t+1}$
    t=t+1
end for
Return S
```

Most real-world multi-objective optimization problems must be constrained. A simple constraint-handling method that uses tournament selection in NSGA-II is proposed. In tournament selection, the picked two solutions from the population may be in three cases: Both solutions are feasible; only one solution is feasible; both solutions are infeasible. When there is no infeasible solution among picked two individuals, the rank is determined by the crowded comparison operator. On the other hand, if both are infeasible, a solution that has a smaller overall constraint violation dominates the other. Always, the infeasible solution is dominated by feasible solutions. More details can be found in [20].

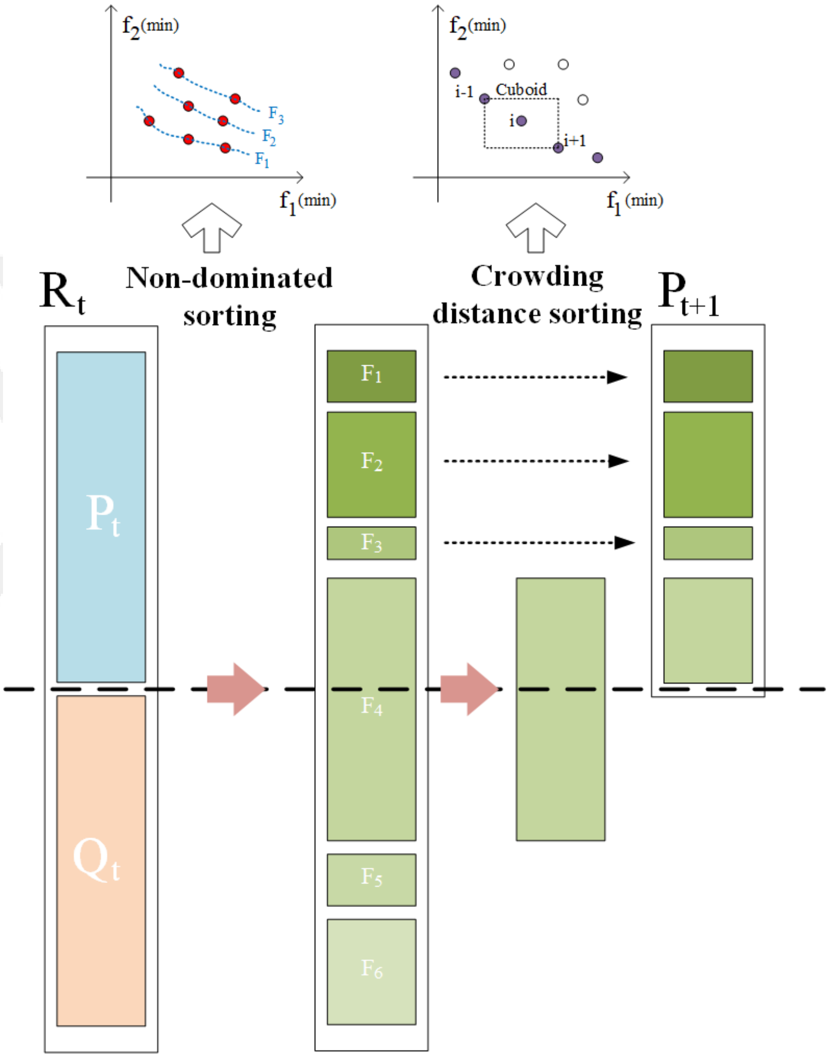


Figure 3.4: General framework of NSGA-II.

3.4. Circuit Modeling: Artificial Neural Network

Artificial Neural Network (ANN) is a mathematical model inspired by biological neural networks consisting of neuron combinations. In a biological neuron, the soma processes information coming from the dendrite and conveys it to the axon. An artificial neuron follows the same procedure as a biological neuron. The body of an artificial neural network sums the weighted input value with a bias and processes it by a mathematical function. The processed information is stored at the output of the neuron. A simple mathematical expression of the artificial neuron model is given in Equation 3.4 where $x_i(k)$, $w_i(k)$, b, F and $y_i(k)$ denotes input value, weight value, bias, transfer function, and output value, respectively.

$$y(k) = F(\sum_{i=0}^{m} w_i(k).x_i(k) + b) \tag{3.4}$$

ANN model can be separated into three basic blocks which are multiplication, summation, and activation. Basic mathematical operations perform the multiplication and summation. However, each weighted input with biases is processed via an activation function such as sigmoid function, exponential linear unit function, rectified linear unit function, etc.

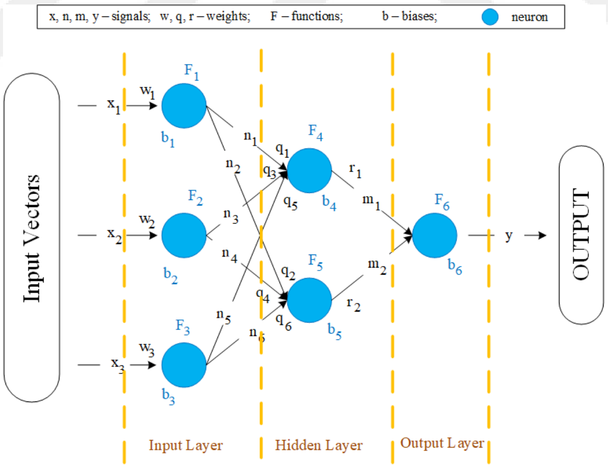


Figure 3.5: A feed-forward artificial neural network.

Most real-world problems cannot be modeled with a single artificial neuron. Artificial neural networks may be constructed with a lot of neurons and several layers for high-accuracy modeling of complex definitions. However, the random connection of the artificial neurons is not preferred due to computing complexity, efficiency, accuracy, and managing consideration. Numerous interconnection ways, also called topology, are presented to address this issue. In this thesis, the circuits have been modeled by a feed-forward artificial neural network approach whose representation is depicted in **Figure 3.5**. As seen in **Figure 3.5**, information flows only in one direction from input to output without back-loops. The input layer containing as many neurons as the number of input vectors, creates new signals by processing weighted and biased input signals. In the hidden layer, each neuron receives information from the signals produced by each neuron in the previous layer. Finally, the output layer processes all formed information through an activation function. The mathematical framework of the aforementioned network example is given in Equation 3.5.

$$\begin{array}{rcc}
n_1 = F_1(w_1.x_1 + b_1) & | & \\
n_2 = F_1(w_1.x_1 + b_1) & | & M_1 = F_4(q_1.n_1 + q_3.n_3 + q_5.n_5 + b_4) \\
. & | & M_2 = F_5(q_2.n_2 + q_4.n_4 + q_6.n_6 + b_5) \\
. & | & \\
. & | & y = F_6(r_1.m_1 + r_2.m_2 + b_6) \\
n_6 = F_3(w_3.x_3 + b_3) & | &
\end{array} \tag{3.5}$$

Basically, an ANN model construction is based on two main phases, which are the neural network development and training of the model. The success of a model depends on the established ANN structure and the training process. Different learning techniques can be used for any ANN topology, depending on the characteristics of the available data. Conventionally, the learning process can be investigated under three sub-titles, which are supervised learning, unsupervised learning, and reinforcement learning. Supervised learning refers to the training procedure, where the ANN model is trained by the training data consisting of input and corresponding output value pairs. The trained model is tested via validation data, which is not included in the learning process. In cases where there is no output information regarding the input vector, the unsupervised learning technique is preferred. On the other hand, reinforcement learning sets the parameters of an ANN without previously provided data. Reinforcement learning guides ANN actions in an environment to maximize reward. In this thesis, circuit models were created through ANN. The models have been trained by the supervised learning technique.

4. EC-BASED ALGORITHM SELECTION

The metaheuristic algorithms can be separated into two main categories: Evolutionary Computation (EC) and Swarm Intelligence (SI), where evolutionary process and behavior pattern on foraging of insects, respectively [57]. There are numerous EC-based and SI-based algorithms have been provided in the literature. Strength Pareto Genetic Algorithm - 2 (SPEA-2)[43], non-dominated Sorting Genetic Algorithm-II (NSGA-II)[20], Multi-objective Evolutionary Algorithm Based on Decomposition (MOEA/D)[58] and Differential Evolution (DE)[59] are well-known EC-based algorithms employed in analog/RF optimization. Particle Swarm (PS)[25], Ant Colony (AC)[28], and Bee Colony (BC)[60] are the most preferred SI-based algorithms. On the other hand, some metaheuristic approaches such as Random Key Genetic Algorithm [61], Bacterial Foraging [62], and Artificial Immune System [63] do not prove the reliability of IC synthesis.

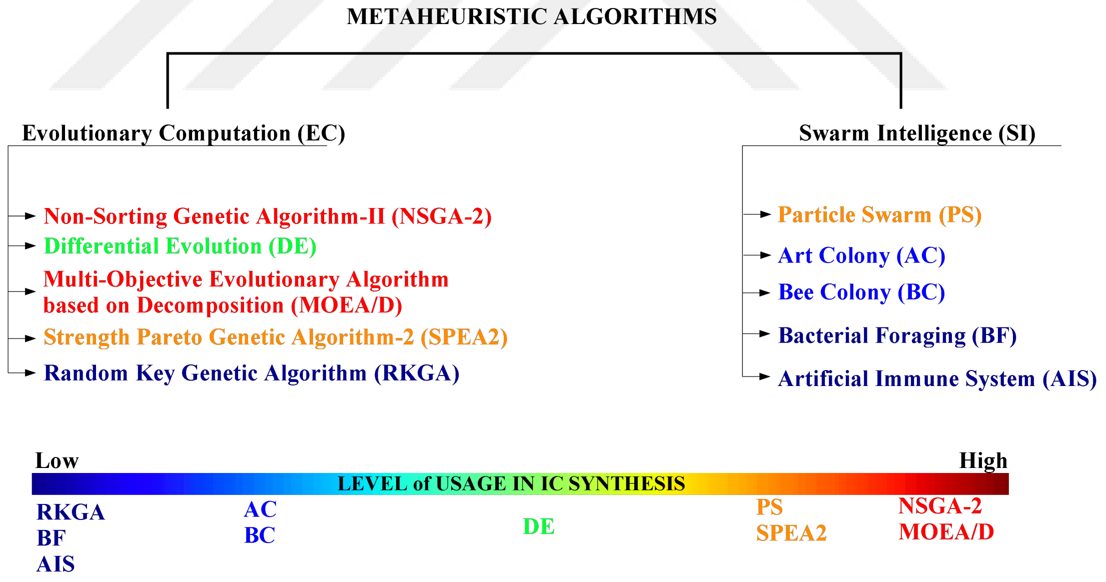

METAHEURISTIC ALGORITHMS
Evolutionary Computation (EC)
Swarm Intelligence (SI)
Non-Sorting Genetic Algorithm-II (NSGA-2)
Differential Evolution (DE)
Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D)
Strength Pareto Genetic Algorithm-2 (SPEA2)
Random Key Genetic Algorithm (RKGA)
Particle Swarm (PS)
Art Colony (AC)
Bee Colony (BC)
Bacterial Foraging (BF)
Artificial Immune System (AIS)
Low
High
LEVEL of USAGE IN IC SYNTHESIS
RKGA
BF
AIS
AC
BC
DE
PS
SPEA2
NSGA-2
MOEA/D

Figure 4.1: Taxonomy of metaheuristic algorithms [64].

Considering the developed automation tools in the literature, EC-based algorithms are frequently preferred to solve IC optimization problems due to their reliability and robustness. A taxonomy of metaheuristic algorithms is shown in **Figure 4.1**. The manual design of analog/RF circuits is tedious since the designer strives to meet design specifications by balancing between enormous trade-offs. Moreover, the IC synthesis process is quite complex due to it contains challenging design objectives with hard design constraints. In this context, although EC-based approaches are pretty suitable for IC synthesis, the algorithm preference is crucial to realize circuit optimization. In this section, the most employed two EC-based approximations, NSGA-II and MOEA/D, have been compared to determine the most convenient algorithm for analog/RF IC sizing. Test benches have been introduced for both analog and RF circuits. The benchmark circuits have been synthesized via a developed multi-objective optimization tool under the same conditions. The algorithm comparison has been realized by the obtained PoF for each circuit with well-known pareto quality metrics [64].

4.1. Analog Benchmark and Synthesis Results

The analog benchmark has been introduced to perform the comparison of NSGA-II and MOEA/D. Five fundamental analog building blocks: Active-loaded differential amplifier, two-stage with compensation, folded cascode OTA, a voltage comparator, and band-gap reference circuits are the members of the introduced benchmark. All synthesis have been performed via the developed tool introduced in chapter 4. The algorithms that were used as optimization kernel were implemented on Python^{TM}, where HSPICE$^{\circledR}$ is utilized as a performance evaluator. All optimizations were carried out on an AMD Ryzen-9 5900X 12-Core Processor with 64 GB RAM. UMC130-nm technology model parameters were included in all circuit netlists.

4.1.1. Active Loaded Differential Amplifier

The conventional active loaded differential amplifier whose schematic is shown in **Figure 4.2** is introduced as the first member of the analog benchmark. The circuit drives 0.5 pF capacitive load, and it is supplied by a single voltage of 1.2V source. Since the amplifier has not been biased symmetrically, the input transistors have been externally biased to ensure that transistors operate in the saturation region. For that reason, the bias voltage is also determined as a design parameter in the decision space. The circuit topology is quite simple, it consists of input differential pairs (M1-M2), active loads (M5-M6), a current mirror (M3-M4), and a resistor used to determine the current.

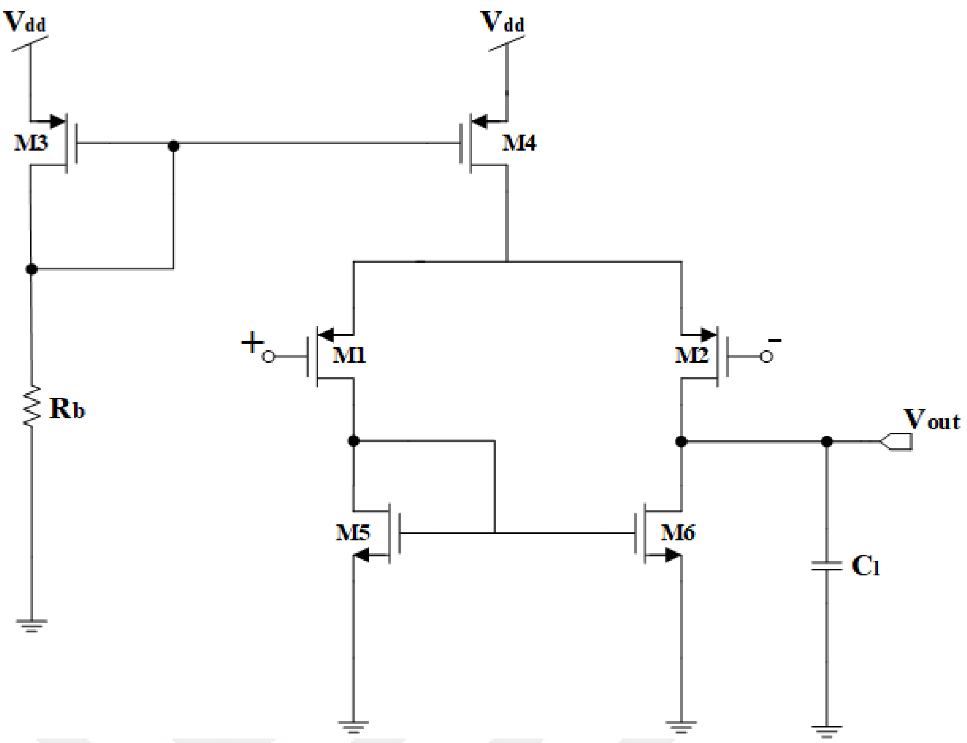


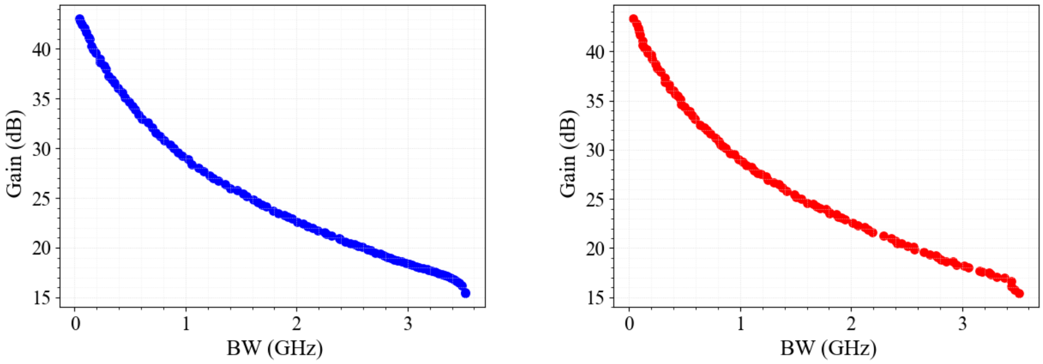
Figure 4.2: Schematic of the active-loaded differential amplifier.

The circuit synthesis has been performed under power consumption, the chip area, and phase margin (PM) constraints, where the DC gain of the amplifier (A_0) and unity-gain bandwidth (f_t) were defined as design objectives. The DC gain was also constrained to prevent premature convergence in first populations. Design parameters, objectives, and constraints for the first analog benchmark member are listed in **Table 4.1**. The population size and generation size were both determined as 100.

Table 4.1: Design parameters, objectives, and constraints for the active-loaded differential amplifier.

Design Parameters					Design Objectives		Design Constraints			
Boundaries	L[μm]	W[μm]	R_b[Ω]	V_{bias}[V]	Gain	BW	Power	PM	Gain	Area
Lower	0.13	1	100	0.2	max	max	$<$ 5mW	$>$ 45°	$>$ 0dB	<300μm^2
Upper	1.3	100	10k	0.8						

The resultant PoFs from the synthesis of the circuit with MOEA/D and NSGA-II for two-objective optimization are shown in **Figure 4.3**. It can be said that the convergence is quite similar and well-distributed PoFs are obtained with both algorithms. Both design spaces have converged to the final solution at the half of generation number.



(a) MOEA/D

(b) NSGA-II

Figure 4.3: PoFs of differential amplifier for MOEA/D and NSGA-II.

4.1.2. Two Stage OTA

The second amplifier in the benchmark is a two-stage OTA given in **Figure 4.4**. The load capacitance is 0.5pF and input transistors have been also biased by external voltage sources because of the absence of a symmetrical supply. The topology is a derivative of the active-loaded differential amplifier, where input pair(M1-M2), load (M3-M4), current mirror (M5-M8), and bias resistor R_b introduce the differential amplifier; M6 is the second stage of the amplifier biased by M7. Since the circuit consists of two cascaded stages, two dominant poles occur that can cause the amplifier to become unstable. For safe operation, the compensation capacitance was inserted between the outputs of the first and second stages to split the dominant poles. Again, while DC gain and unity gain frequency were design objectives; phase margin, power consumption, chip area, and DC gain were determined as design constraints. Decision space boundaries, objectives, and constraints are tabulated in **Table 4.2**. Unlike the differential amplifier, the compensation capacitor has been added to the design space, and power consumption has been constrained by 10 mW.

Table 4.2: Design parameters, objectives, and constraints for two-stage OTA.

Design Parameters						Design Objectives		Design Constraints			
Boundaries	L[μm]	W[μm]	R_b[Ω]	V_{bias}[V]	C_c[pf]	Gain	BW	Power	PM	Gain	Area
Lower	0.13	1	100	0.2	0.1	max	max	<10mW	$>45°$	>0	<1000m^2
Upper	1.3	100	10k	0.8	50						

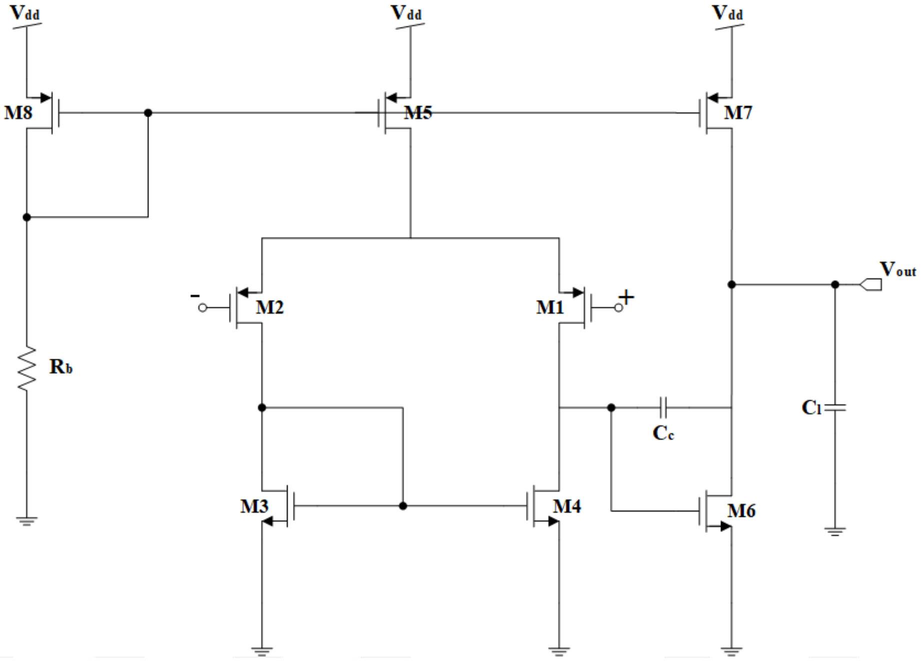
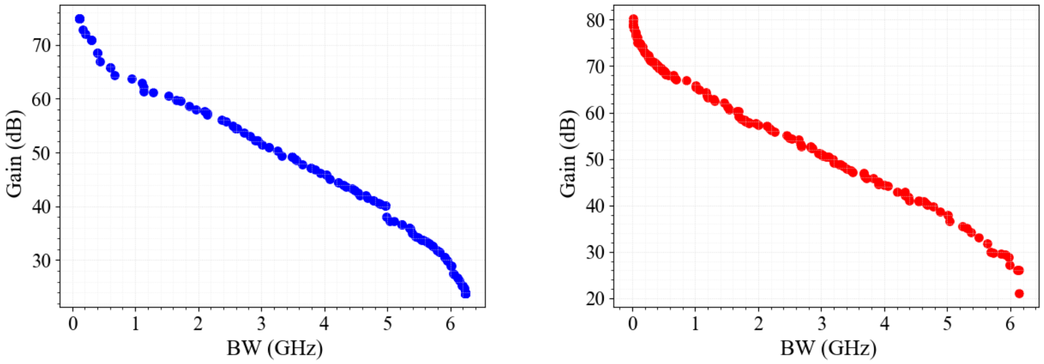


Figure 4.4: Schematic of the two-stage OTA.



(a) MOEA/D

(b) NSGA-II

Figure 4.5: PoFs of two-stage OTA for MOEA/D and NSGA-II.

4.1.3. Folded Cascode OTA

Another amplifier circuit, folded cascode OTA has been selected as the third benchmark member. The circuit drives the same capacitive load and is supplied by a single power supply. In addition, the differential input pair was biased to ensure keeping in the saturation region. The circuit schematic is shown in **Figure 4.6**. The current created by the cascode current source is steered by M9 and M10 to the output stage, and M4 sinks some portion of the current of M9 and M10 to bias the input pair. The amplifier's output stage consists of a stack of cascaded transistors, some of which need proper gate biasing.

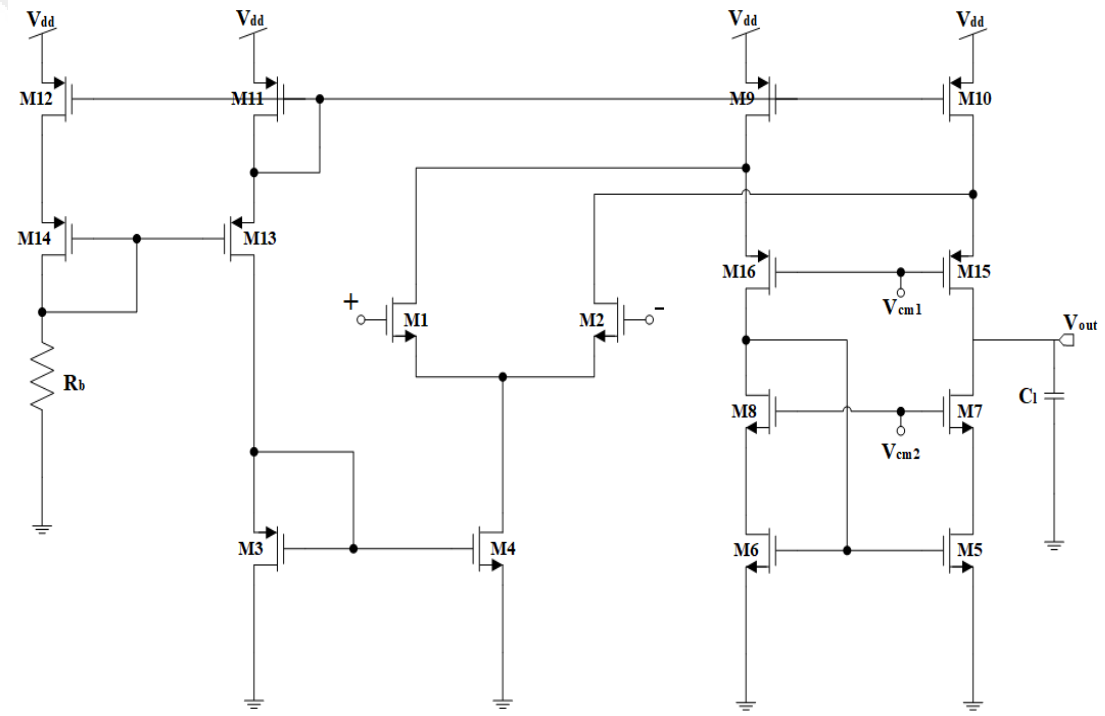
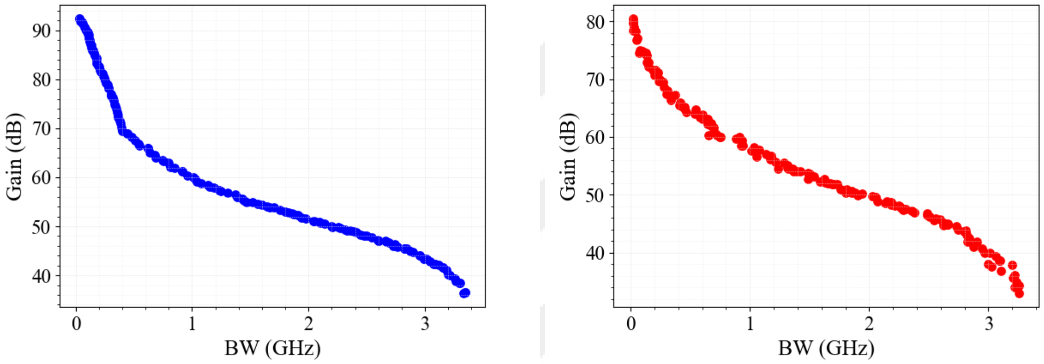


Figure 4.6: Schematic of the folded cascode OTA.

Table 4.3: Design parameters, objectives, and constraints for folded cascode OTA.

	Design Parameters						Design Objectives		Design Constraints			
	L[μm]	W[μm]	R_b[Ω]	V_{bias}[V]	V_{cm1}[V]	V_{cm2}[V]	Gain	BW	Power	PM	Gain	Area
Min	0.13	1	100	0.2	0.1	0.2	max	max	<10mW	$> 45°$	> 0	<1000umm^2
Max	1.3	100	10k	0.8	0.8	1						

In the folded cascode OTA optimization, the design objectives and design constraints are the same as in the two-stage synthesis. On the other hand, the compensation capacitor has been removed from the design parameter, and common mode voltage bias values have been added to the decision space. Design parameters, objective space functions, and optimization constraints are provided in **Table 4.3**. The synthesis has been carried out by 400 generations with 150 population size. Considering the resultant PoFs are given in **Figure 4.7**, the MOEA/D is better than NSGA-II in terms of design space coverage and spreading of solutions. Moreover, MOEA/D has converged to the final PoF with 225 iterations while no significant convergence improvement can be mentioned after 290 iterations in NSGA-II.



(a) MOEA/D

(b) NSGA-II

Figure 4.7: PoFs of folded cascode OTA for MOEA/D and NSGA-II.

4.1.4. Voltage Comparator

The voltage comparator circuit which is widely used in many analog applications was selected as another analog benchmark circuit. The latched comparator shown in **Figure 4.8** roughly consists of three parts: Active-loaded amplifier which is introduced by the transistors M2, M3, M5, and M6; the cross-coupled transistors that boost the differential gain and balance the output resistance; a common source amplifier by M8; and an inverter that gives the final state of the comparison signal. The circuit is symmetrically supplied with 0.9V voltage sources, where the bias voltage of the M4 transistor is constant 0.1V.

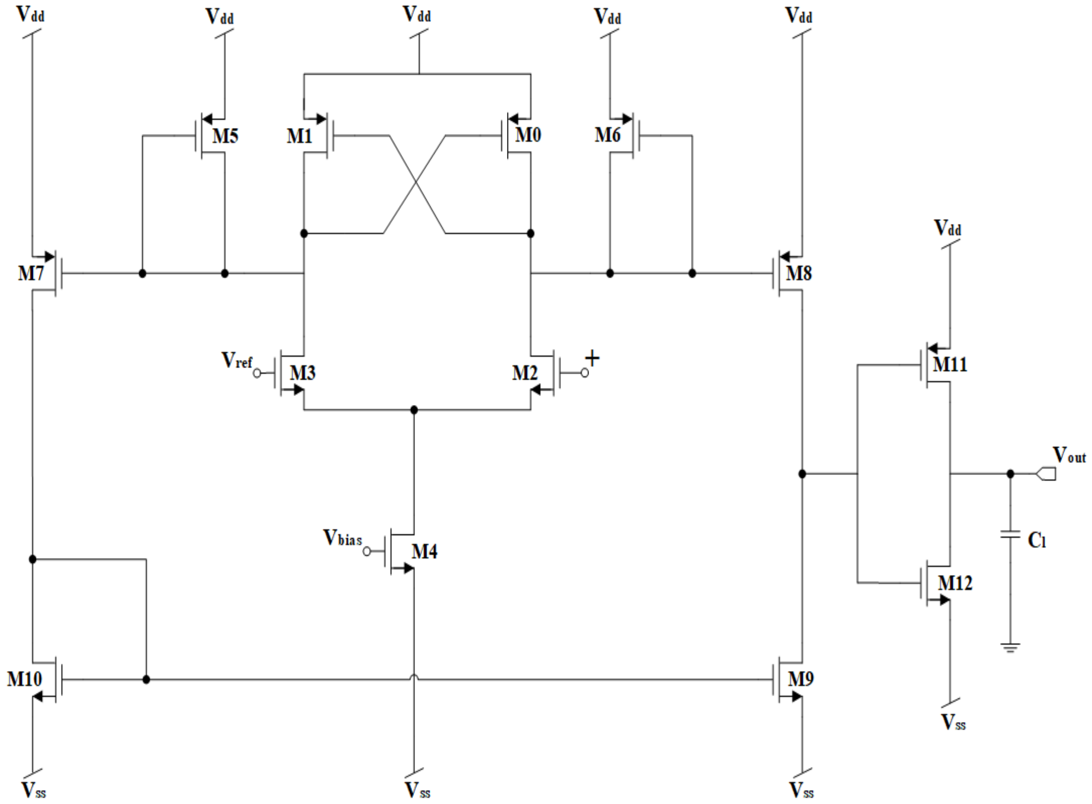


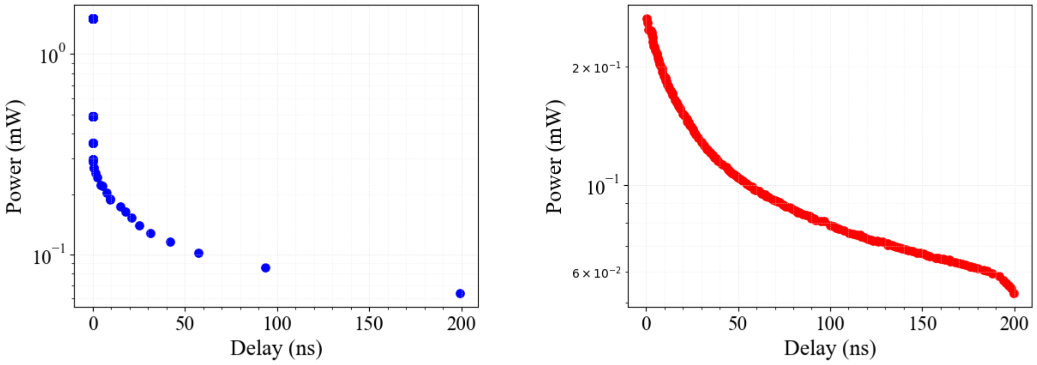
Figure 4.8: Schematic of the voltage comparator.

Power consumption and delay have been defined as both design goals and design constraints for the comparator circuit. The time delay has been intentionally constrained to eliminate infeasible solutions. Decision space boundaries and other optimization parameters are listed in **Table 4.4**. The circuit has been synthesized with 60 generations each containing 150 individuals. All transistor lengths have the same size to achieve high speed. Furthermore, the design space boundaries have been narrowed to facilitate the optimization process.

Table 4.4: Design parameters, objectives, and constraints for voltage comparator.

Design Parameters			Design Objectives		Design Constraints	
	L[μm]	W[μm]	Power	Delay	Power	Delay
Max	0.13	0.16	min	min	< 5mW	>0
Min	1.3	80				

As seen from the optimization results given in **Figure 4.9**, a well-distributed pareto optimal front was achieved by NSGA-II, however, MOEA/D has stacked at local minima which is not desired situation.



(a) MOEA/D

(b) NSGA-II

Figure 4.9: PoFs of voltage comparator for MOEA/D and NSGA-II.

4.1.5. Bandgap Reference

Bandgap reference (BGR) is one of the most critical analog sub-block which is necessary in any system-level design to avoid being affected by changing temperature conditions. The well-known Banba's BGR circuit [65] has been considered as the final member of the analog benchmark. The circuit whose schematic is presented in **Figure 4.10** is supplied by a single voltage source of 1.8V. The presented BGR is formed of the start-up circuit, current source MOSFETs, and a kind of resistor-diode combination to perform temperature compensation. The error amplifier is utilized to provide equal voltage at both two emitter terminals. The start-up circuit (MS1-MS2-MS3-MS4) prevents DC errors that may occur at the initializing phase. The resistor-diode combination refers to the R0, R1, R2, R3, Q1, and Q2 configuration, where Proportional to Absolute Temperature (PTAT) current flows through R0, and Complementary to Absolute Temperature (CTAT) current flows through R2. Thus, a temperature-independent current is generated at the output.

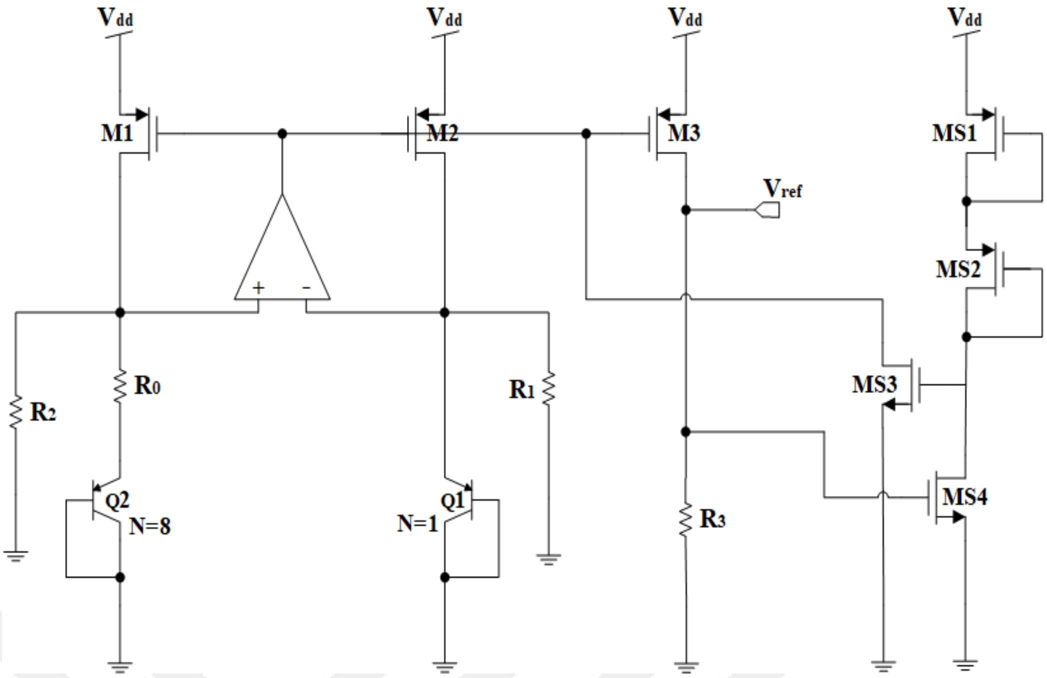


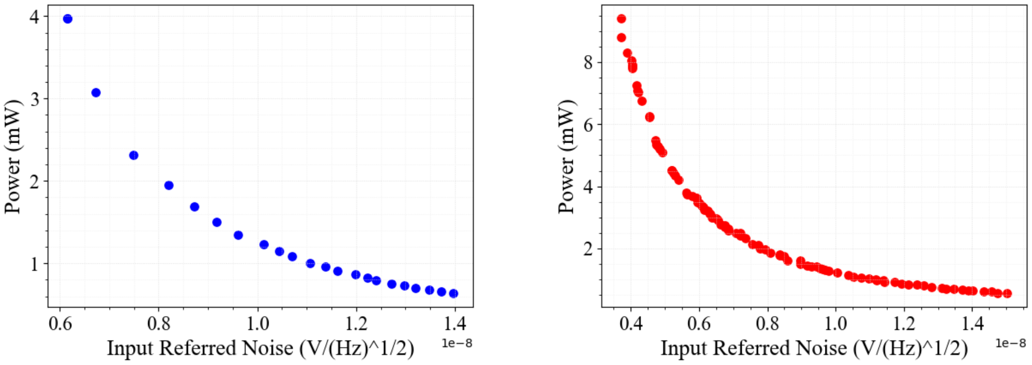
Figure 4.10: Schematic of the BGR circuit.

The input referred noise and power consumption were determined as objective functions. On the other hand, power consumption, temperature coefficient, and the output reference voltage introduce the constraints of the optimization process. Design parameters, objectives, and constraints are listed in **Table 4.5**.

Table 4.5: Design parameters, objectives, and constraints for BGR.

Design Parameters				Design Objectives		Design Constraints		
Boundaries	L[μm]	W[μm]	R[Ω]	Power	Noise	TC	Power	V_{out}
Lower	0.13	0.65	100	min	min	<100 ppm C°	<10 mW	0.29V < V_{ref} < 0.31V
Upper	1.3	97.5	100k					

The multi-objective optimization of BGR has been carried out by 350 generations with 150 population size. The resultant PoFs are depicted in **Figure 4.11**, where MOEA/D has not found optimum solution points and it has created rough PoF, while well-distributed PoF obtained via NSGA-II after 290 iterations.



(a) MOEA/D

(b) NSGA-II

Figure 4.11: PoFs of bandgap reference for MOEA/D and NSGA-II.

4.2. RF Benchmark and Synthesis Results

RF circuit synthesis is more problematic, since they include passive elements such as inductors and capacitors, so the RF circuits require more sophisticated performance analyses, like steady-state and harmonic balance analyses. Generally, the performance of RF circuits is evaluated by EM simulations to achieve highly accurate estimation. However, even if EM simulation is highly accurate, it is not efficient in terms of simulation time. On the other hand, performance evaluation using ideal models for passive devices gives optimistic results, where layout parasitic effects are not considered. To increase the accuracy with promising acceptable evaluation time, physical-based parasitic aware models are utilized for passive elements as proposed in [42], where the layouts and equivalent representations of the MIM (Metal-Insulator-Metal) capacitor and planar inductor are shown in **Figure 4.12**. The metal width (W), the length (L), and well spacing (D_x and D_y) constitute capacitor design objectives, while '$2-\pi$' model is utilized for inductor modeling where metal width, the number of turns, the metal spacing, the outer diameter, and the well spacing denoted as W, N_t, D_o, s, and $D_{x,y}$, respectively. Corresponding equivalent circuits of each model were introduced in the circuit netlist as sub-circuit. The RF benchmark includes two circuits: A cross-coupled LC CMOS oscillator, and a MOS cascode low-noise amplifier (LNA). Circuit syntheses have been carried out by the developed tool, and HSPICERF$^{®}$ has been employed as the performance evaluator.

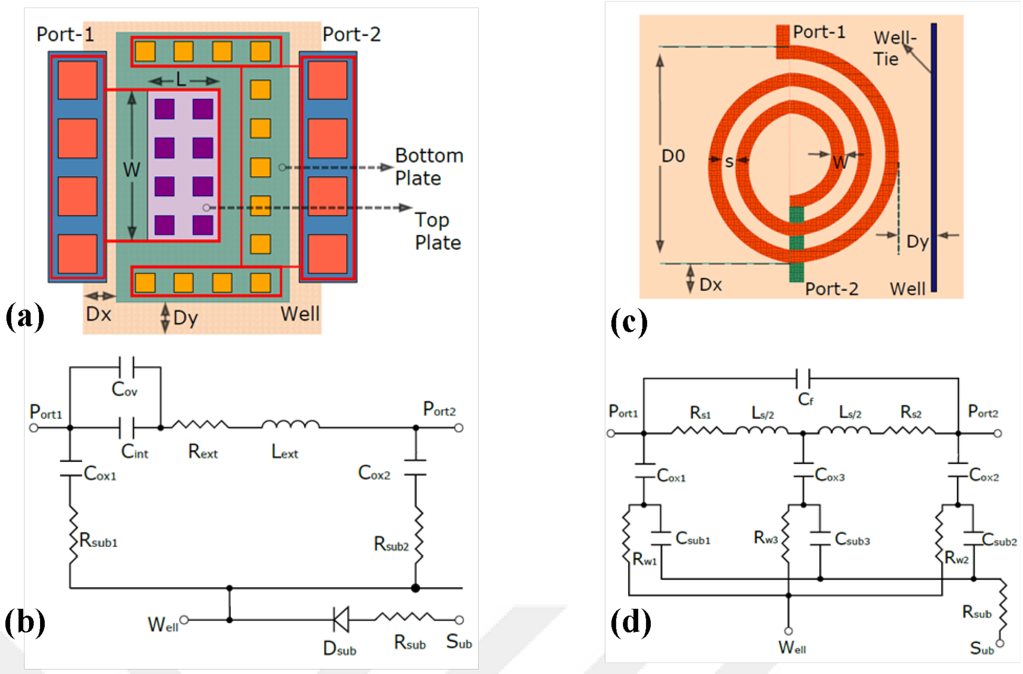


Figure 4.12: MIM capacitor and planner inductor layouts [42].

4.2.1. CMOS Differential Cross-coupled LC Oscillator

The cross-coupled LC oscillator whose schematic is shown in **Figure 4.13** was considered as the first member of the proposed RF benchmark. While the LC tank is the resonator part of the circuit, cross-coupled transistors generate the required negative impedance to satisfy Barkhausen criteria. Owing to the presence of complementary cross-coupled pairs, the power consumption is reduced when compared to NMOS-only or PMOS-only pairs. The circuit is biased by a symmetrical supply voltage of 1.2V The initial condition statement has been determined at the output node as 0.6V to avoid initializing problems and to ensure proper operation. In oscillator optimization, the phase noise (PN) and the power consumption were determined as both design objectives and design constraints. Although the phase noise should be minimized as much as possible, an inequality constraint was determined at the extreme minimum point which introduces a boundary to prevent infeasible solutions. In addition, the oscillation frequency, oscillation amplitude, and power consumption were also considered as design constraints. The design objectives and constraints are listed in **Table 4.6**. Here, it should be emphasized that the oscillation frequency has been constrained as narrow as possible to achieve a precise targeted frequency value. In addition, under-damped solutions are eliminated by checking the oscillation amplitudes at 3-4ns and 7-8ns time intervals.

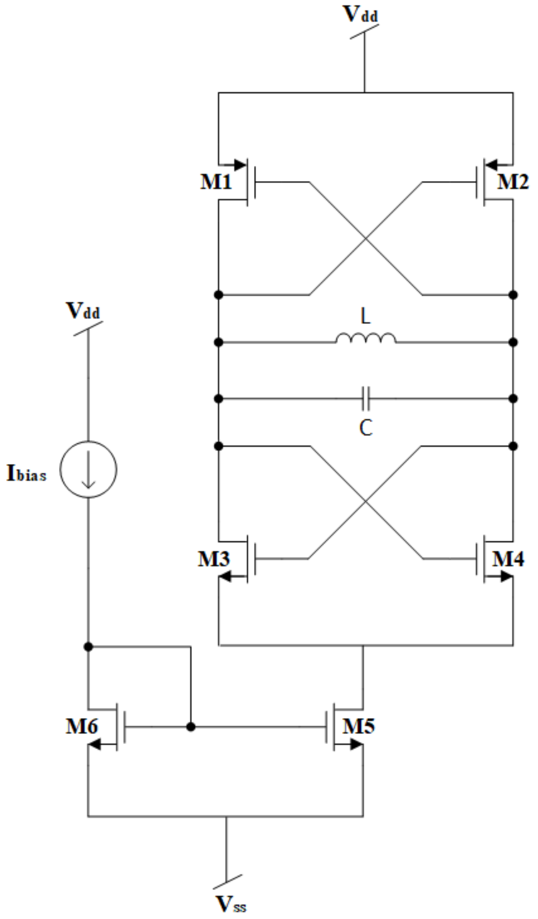


Figure 4.13: Schematic of the CMOS differential cross-coupled LC oscillator circuit.

Table 4.6: The design objectives and the constraints of CMOS oscillator.

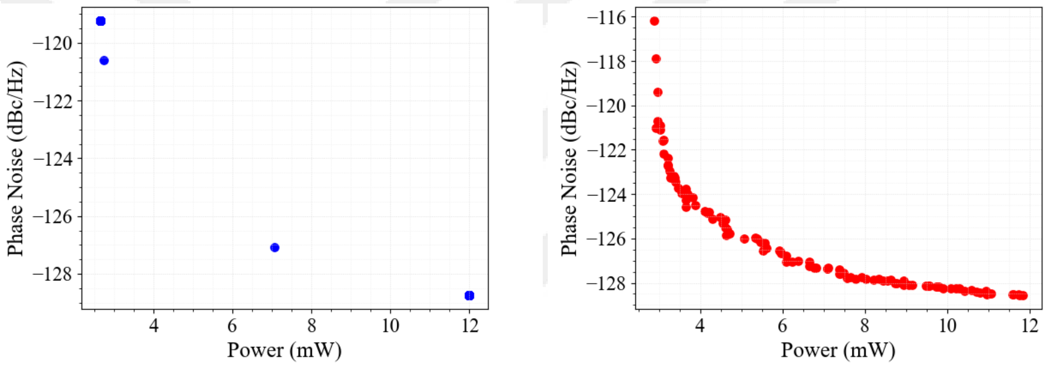
Constraints	Values
* Phase Noise @ 1MHz	-150[dBc/Hz] <PN <-115 [dBc/Hz]
* Power Consumption	<10mW
Oscillation Frequency	2.39 GHz $<f_o<$2.41 GHz
Oscillation Peak Amplitude @3ns and @7ns	>1V

* Phase Noise and Power Consumption are also design objectives to be minimized.

The population size was determined as 100, and the maximum generation number was determined as 400 which introduces stopping criteria for the optimization. The design parameters are given in **Table 4.7**. Even if the number of design parameters seems to be less than the previously synthesized analog circuits, the automatic synthesis of the oscillator is highly problematic due to its highly non-linear nature and simulation/measurement difficulties. To ensure accurate measurement results via SPICE evaluation, a suitable test bench should be constructed.

Table 4.7: Design parameters for CMOS oscillator.

Boundary	L[μm]	$W_{1:4}$[μm]	$W_{5:6}$[μm]	I_{bias}[mA]	W_{cap}[μm]	L_{cap}[μm]	D_o[μm]	W_{ind}[μm]	N_t	s[μm]
Lower	0.13	10	10	0.5	10	10	135	3	5	1.5
Upper	1	35	500	6	30	35	150	5.5	7.5	2.5



(a) MOEA/D

(b) NSGA-II

Figure 4.14: PoFs of oscillator for MOEA/D and NSGA-II.

The obtained PoFs for the oscillator circuit are depicted in **Figure 4.14**. Considering the figure, it can be interpreted that NSGA-II succeeds in generating a proper front, whereas MOEA/D gives a few solutions in the same trajectory.

4.2.2. Cascode Low-noise Amplifier

The last of the synthesized circuit, the 2.4GHz LNA circuit in the cascode topology, has been chosen as the second member of the RF benchmark. The schematic of the circuit is given in **Figure 4.15**, where the V_{dd} value is 2.4V. The L2 and parasitic capacitance occurred at the drain of M2 determine the resonance frequency of the circuit. M3 is a current sink that steers the generated current by M1.

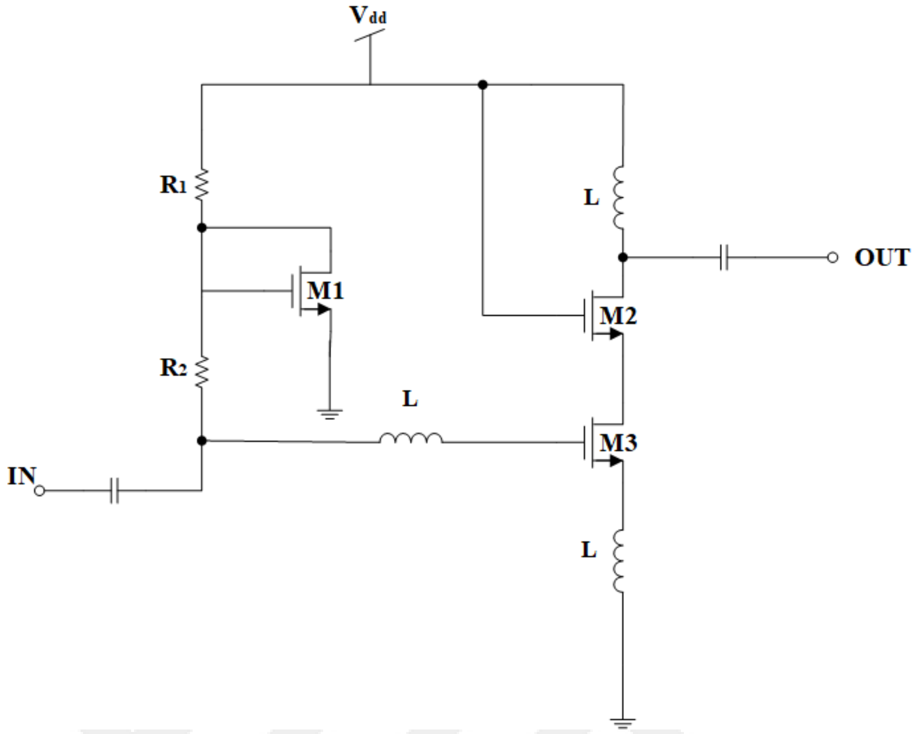


Figure 4.15: Schematic of the cascode LNA circuit.

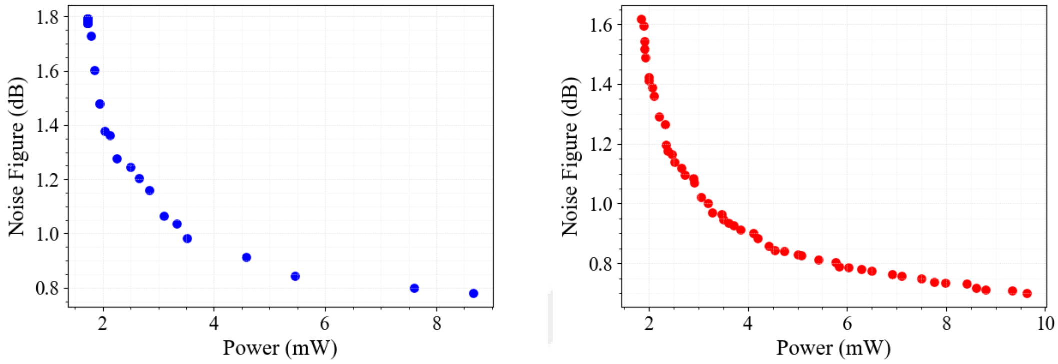
Table 4.8: Design Parameters of cascode LNA

Boundaries	L[μm]	W[μm]	R_1[Ω]	R_2[Ω]	D_o[μm]	W_{ind}[μm]	N_t	s[μm]
Lower	0.12	24	1k	1k	75	2	2.5	1.5
Upper	1	480	15k	15k	150	10	7.5	2.5

The physical-based equivalent circuit model has been placed into the circuit netlist to realize parasitic-induced inductor modeling for all inductors, similar to the oscillator circuit. The design parameters and decision space boundaries are listed in **Table 4.8**. On the other side, the circuit was optimized within 2-dimension objective space, where the noise figure (NF) and power consumption are objective functions, under S-parameters constraints. To limit the power, the power consumption was also determined as a design constraint besides S-parameters constraints. Design objectives and constraints are tabulated in **Table 4.9**. The virtual matching networks are used to match input and output impedances to 50 Ω in the simulation environment.

Table 4.9: The design objectives and the constraints of LNA.

Objective Space		Constraints			
Noise Figure	Power Consumption	Power [mW]	S_{11} [dB]	S_{12} [dB]	S_{21} [dB]
min	min	<10	<-15	<-30	>10



(a) MOEA/D

(b) NSGA-II

Figure 4.16: PoFs of LNA for MOEA/D and NSGA-II.

The circuit has been synthesized with 500 generations for 50 population sizes. Considering resultant PoFs, NSGA-II has created well-distributed solutions, while MOEA/D has reached a poor quality PoF, where convergence and continuity are not as good as the PoF obtained by NSGA-II. Both circuits achieved the final PoF in 190 iterations.

4.3. Discussion and Comparison of the Synthesis Results

In the previous sections, benchmark circuits have been synthesized through the developed tool. In an optimization process, the searching algorithm seeks to possible best solutions with satisfying determined constraints. Therefore, the best way of determining searching algorithm effectiveness is PoF quality evaluation via PoF quality metrics. In this context, performances of MOEA/D and NSGA-II have been compared by well-known five different PoF quality metrics which are Generational Distance (GD), Inverted Generational Distance (IGD), Spacing (SP), Spread (Δ), and Maximum Spread (MS). The GD is about how converge the solutions are to the real solution set. As seen from the Equation 4.1, it determines the average distance between each solution revealed in the obtained PoF and its closest neighbor solution in true PoF. On the other side, even if IGD is given in Equation 4.2 looks quite similar to GD, it is the inverse of GD. The IGD is utilized to determine how far the true PoF solutions are from evolved solutions, where the evolved solution is the reference for real PoF. These two metrics can be classified into two indicator categories: Convergence and distribution, and convergence.

$$GD(Y_N;Y_P) = \frac{1}{Y_N}\left(\sum_{y^1\in Y_N}\underset{y^2\in Y_P}{min}\left\|y^1-y^2\right\|^p\right)^{\frac{1}{p}} \tag{4.1}$$

$$IGD(Y_N;Y_P) = \frac{1}{Y_P}\left(\sum_{y^2\in Y_P}\underset{y^1\in Y_N}{min}\left\|y^1-y^2\right\|^p\right)^{\frac{1}{p}} \tag{4.2}$$

The spacing indicator is utilized to determine the homogeneity of the obtained solutions which gives the distribution of the solution in the evolved PoF. The SP metric is calculated byEquation 4.3, where $d^1(y^j, Y_N\backslash\{y^j\})$ is the Manhattan distance between y_j and its closest neighbor in the solution set Y_N, and $\bar{d}$ means of all calculated Manhattan Distance values.

$$SP(Y_N) = \sqrt{\frac{1}{|Y_N-1|}\sum_{j=1}^{|Y_N|}\Big(\bar{d}-d^1(y^j,Y_N\backslash\{y^j\})\Big)^2} \tag{4.3}$$

Another indicator, Δ, classified into dispersion and range, determines the extent of propagation solutions relative to the actual solution set. Metric Δ is calculated by Equation 4.4, where $d^c(y^j, Y_N\backslash\{y^j\})$ gives Euclidean distance between two consecutive solutions of the obtained pareto front Y_N, and $\bar{d^c}$ denotes the mean value of the $d^c(y^j, Y_N\backslash\{y^j\})$. Here, it is worth noting that the indicator can be misunderstood if pareto front is formed by separated multiple solution groups. Since it utilizes $l2$ norm for the determined distance between each solution and its closest neighbor. Also, $min_{y\in Y_N}\left\|y^{i,*}-y\right\|$ for i=1,2 denotes the $l2$ distance between two extreme solution points of real PoF and corresponding boundary solutions of the obtained solution space.

$$\Delta(Y_N;Y_P) = \frac{\sum_{i=1}^{2} \underset{y\in Y_N}{min} \|y^{i,*}-y\| + \sum_{j=1}^{|Y_N|-1} \left| d^c(y^j, Y_N\backslash\{y^j\}) \right| - \bar{d}^c}{\sum_{i=1}^{2} \underset{y\in Y_N}{min} \|y^{k,*}-y\| + (|Y_N|-1)\bar{d}^c} \tag{4.4}$$

The MS metric measures how well the resultant pareto approximation covers the true pareto front. It can be calculated by Equation 4.5, where $PF^{max,min}_{y^i\in Y_N}$ and $PF^{max,min}_{y^i\in Y_P}$ are the maximum or minimum value of ith objective in the obtained solution set and real PoF, respectively. In addition, M represents the objective number. Furthermore, information about performance metrics can be found in [66], [67].

$$MS = \sqrt{\frac{1}{M}\sum_{i=1}^{M}\left[\frac{min(PF^{max}_{y^i\in Y_N}, PF^{max}_{y^i\in Y_P}) - max(PF^{min}_{y^i\in Y_N}, PF^{min}_{y^i\in Y_P})}{PF^{max}_{y^i\in Y_P}, PF^{min}_{y^i\in Y_P}}\right]^2} \tag{4.5}$$

If high-quality PoF is to be mentioned, the metrics GD, IGD, Δ, and SP should be as small as possible, on the other side, MS should be as large as possible. To evaluate the obtained PoF quality via these metrics, a proper PoF is necessary. To prevent any misleading results in the PoF quality evaluation process, a logical operator called Entia has been defined, which gives 1 if proper PoF is obtained and vice versa. In addition, the convergence point (CP) is defined to indicate the number of iterations at which the algorithm converges to the solution. Eventually, a figure of merit (FoM) in Equation 4.6 formed by combining all these measures have been defined to do an overall evaluation.

$$FOM = \frac{MS \times \text{Entia}}{GD \times IGD \times S \times \Delta \times CP} \tag{4.6}$$

Most of the performance indicators require a true PoF to evaluate the quality of the resulting PoF. In this context, to obtain true PoF for provided benchmark circuits, all circuits have been synthesized with increasing maximum iteration count 4 times larger than their nominal values. To quickly assess the performance of the algorithm, all of the metric indicator results, Entia functions, and figure of merit calculations for both algorithms are tabulated in **Table 4.10**.

Table 4.10: The comparison table of competitor algorithms.

	Differential Amplifier		Two-Stage OTA		Folded Cascode OTA		Voltage Conparator	
	MOEA/D	NSGA-II	MOEA/D	NSGA-II	MOEA/D	NSGA-II	MOEA/D	NSGA-II
GD[↓]	**0.01**	0.017	0.05	0.05	0.14	**0.07**	1	**0.01**
IGD[↓]	0.03	0.03	0.19	**0.14**	**0.14**	0.16	0.94	**0.02**
SP[↓]	0.07	**0.06**	0.20	**0.18**	**0.07**	0.09	1	**0.06**
Δ[↓]	**0.13**	0.14	0.51	**0.38**	**0.26**	0.28	0.84	**0.22**
MS[↑]	0.99	0.99	0.93	**0.98**	**0.98**	0.96	0.97	0.97
Entia [0/1]	1	1	1	1	1	1	0	1
CP	50	50	185	**180**	**225**	290	X	**20**
FoM	7.25e3	**7.85e3**	5.18	**11.37**	**12.21**	11.72	0	**18.37e3**

	Band-gap Reference		Low-Noise Amplifier		CMOS Oscillator	
	MOEA/D	NSGA-II	MOEA/D	NSGAII	MOEA/D	NSGA-II
GD[↓]	**0.01**	0.02	**0.02**	0.04	0.04	**0.03**
IGD[↓]	1.59	**0.06**	1	**0.10**	0.37	**0.08**
SP[↓]	0.36	**0.19**	0.87	**0.08**	0.59	**0.17**
Δ[↓]	0.52	**0.46**	1	**0.34**	0.69	**0.31**
MS[↑]	0.56	**0.99**	**0.99**	0.97	0.87	**0.94**
Entia [0/1]	0	1	0	1	1	1
CP	X	**290**	X	**300**	**150**	190
FoM	0	**32.54**	0	**29.71**	0.96	**39.11**

Considering the comparison summary, MOEA/D is slightly better than NSGA-II in obtaining well-distributed true PoF for the synthesis of the differential amplifier and folded cascode OTA circuits. As with the other two-amplifier syntheses, both algorithms perform head-to-head in the two-stage OTA synthesis, where NSGA-II is slightly better than MOEA/D. In particular, MOEA/D perform well in solving relatively simple circuit sizing problem with the help of decomposition operation. On the other side, NSGA-II provides homogeneously distributed solutions as seen from spacing metric results. These well-distributed solutions in NSGA-II are obtained owing to the crowding distance operator which ensures diversity in the solution set. Although MOEA/D has not reached the proper distributed PoF for the voltage comparator, the spread metric demonstrates a commendable level of efficacy since the algorithm has found extreme solution points in the objective space. As mentioned before, only one metric does not make sense on its own. This situation has been proved by indicator results for MOEA/D for the voltage comparator circuit. While the spread metric is good enough, the other metrics of MOEA/D cannot even be compared to those of NSGA-II. The NSGA-II outperforms MOEA/D for the fourth benchmark circuit. In addition, in BGR, the GD metric result can lead to misunderstanding MOEA/D performance on band-gap reference circuit synthesis. Some performance measurements produced misleading results for the oscillator's PoF. Even though MOEA/D could not achieve uniform pareto optimal solutions, the algorithm looks better in terms of GD. It is caused by the GD giving limited information about distribution when the solutions spread into multiple clusters and some holes are formed in the resultant PoF. Also, MOEA/D has found extreme solutions, thus the algorithm is better than NSGA-II in terms of maximum spread metric. However, there cannot be any doubt that NSGA-II is superior to MOEA/D in oscillator synthesis. In the synthesis of the LNA circuit, even though MOEA/D has created proper pareto optimal solutions, NSGA-II outperforms MOEA/D in terms of all performance metrics. In summary, it can be concluded that the efficacy of the MOEA/D algorithm diminished with increasing circuit complexity. The author considers that the cause of this situation can be explained by two main reasons: One possible reason that might be the decomposition operator causes reduced diversity in the population, especially in the synthesis of more complex circuits, which may constitute many infeasible solutions in first generations; the second one can be the applied constraint handling method may be insufficient for circuit optimization. In conclusion, it has been seen that NSGA-II can solve multi-objective analog/RF sizing problems even if the synthesized circuit is complex or strictly constrained.

5. ANN MODEL GENERATION AND ANN-BASED OPTIMIZATION

To accelerate the optimization process, especially for highly complex circuit synthesis, the ANN models have been created. The ANN model generation flowchart for analog/RF sub-blocks is given in **Figure 5.1**. The synthesis process begins as in previous sections with SPICE-based optimization, where NSGA-II searches best solutions in the objective space. After initializing optimization, ANN model training starts with the first created population. The design objectives, constraints, and parameters that constitute optimization solutions and decision spaces are utilized to model training. The SPICE-based synthesis and model training have been performed simultaneously until the training process is accomplished. After that, the trained model then replaces SPICE to perform the performance evaluation task, then the synthesis process turns into machine learning (ML)-based optimization. Determining the stopping criteria for the model training is quite critical, both efficiency and accuracy must be considered at the same time. Therefore, a condition is defined by the user to finish model training.

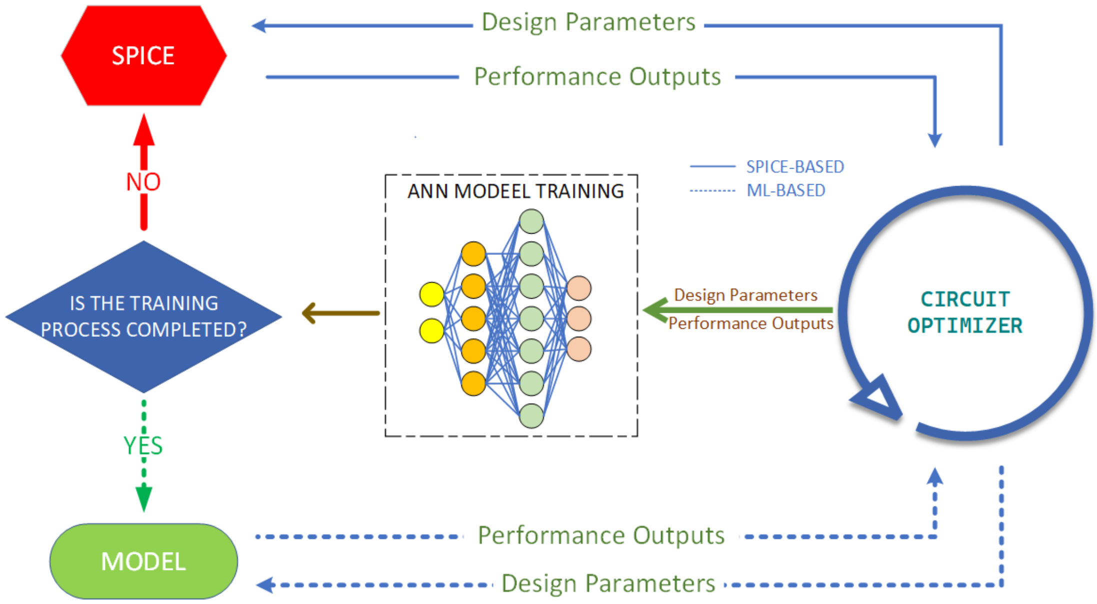


Figure 5.1: Flowchart of ML-based optimization tool.

The model training phase plays a very critical role in providing both efficiency and accuracy which are two crucial elements of an optimization tool. If the model is trained only using feasible solutions, especially for complex circuits, the vision of the trained model may narrowed causes to perform inaccurately. On the other side, when the model is trained by a huge dataset to increase accuracy, the synthesis time extends beyond practical time limits. In this regard, two important points should be taken into consideration for model training in developed tools: Including both infeasible and feasible solutions in training data, and determining proper stopping criteria without sacrificing accuracy or efficiency.

5.1. Data-set Creation and Benchmark Examples

The ANN modeling consists of two phases which are data-set creation and model training. Both of these have the same importance for the construction of a high-accuracy model. The inadequacy of one can render an ineffective model. The data set creation process is well established for some ANN modeling, especially image modeling and classifying, where data can be easily expressed mathematically. However, modeling some real-world examples may be quite tedious since it requires a burdensome dataset-generation process. In this context, as the circuit performance highly depends on the provided technology node, and it changes with the alteration of environmental effects such as bias voltage, load impedance, and temperature, it is impossible to reach the available dataset to any analog/RF topology for modeling. The developer has to generate and organize the training dataset, manually. To collect data that will be added dataset structure, SPICE simulations are necessary. Although there are many different data collection methods for analog/RF circuit training, the data creation via parametric sweep in determined parameter interval is frequently used to obtain training data. Changing any design parameter means a new design, which can be included in the training dataset. Even though this method is simple, it contains a trade-off between accuracy and efficiency. A large number of simulation results are required to comprehensively explore the entire design space, leading to a decrease in the efficiency of the training process. On the other side, when the dataset is narrowed, over-fitting may occur due to inadequate sample size.

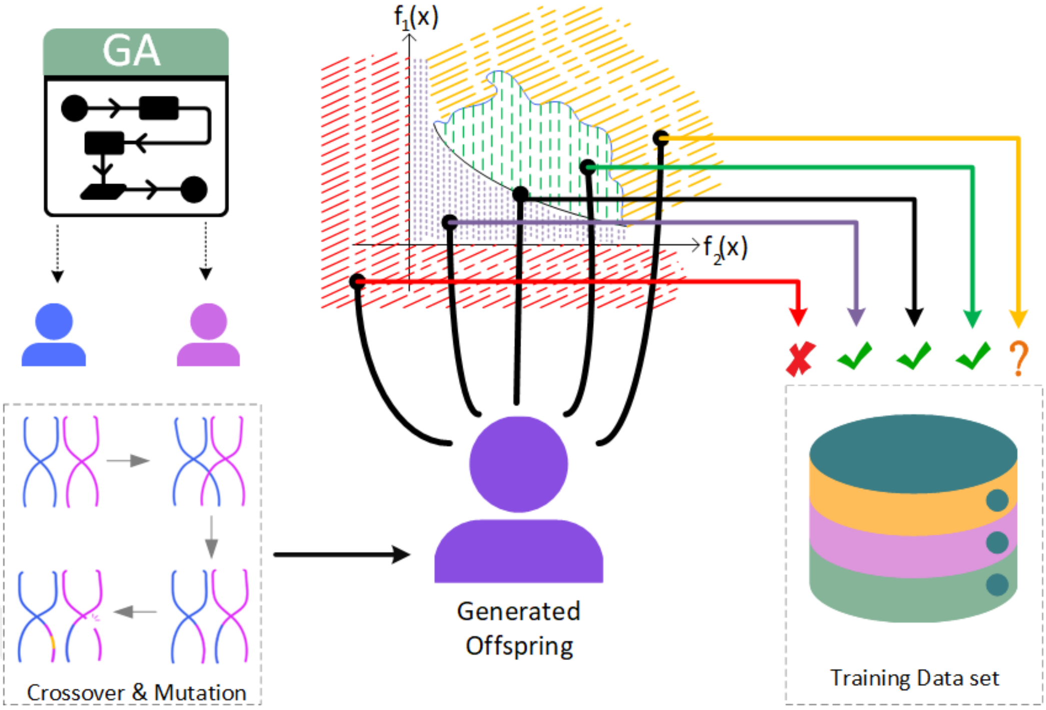


Figure 5.2: An illustration of dataset generation in developed tool.

To overcome this bottleneck, training data are generated by the genetic algorithm, which helps to reach whole possible solutions in the design space of a particular circuit. The genetic algorithm ensures diversity in evolving populations with genetic operations. Thus, diversified sampling is guaranteed without sacrificing time consumption. The dataset generation process of the developed ML-based optimization tool is illustrated in **Figure 5.2**. The offspring is generated using the genetic information of their parents, owing to the genetic algorithm that applies crossover and mutation on parents' chromosomes. Let us think about a multi-objective optimization problem that has two objective functions and both of them should be minimized. Here, generated offspring may belong to five distinctive regions in the objective space which are the feasible solution region, pareto optimal front line, two infeasible solution regions that include better or worse candidates in terms of the only objective function, and the infeasible solution area out of functions boundaries. The solutions belonging to the feasible solution region are directly joined to the training dataset. On the other side, in cases where the solution is infeasible, it is decided whether to include it in the data set or not, according to the location of the solution in the objective space. If the infeasible solution has better objective values than pareto optimal solutions, that individual is directly included in training data. Even if, in most optimization cases, it is rare to obtain an individual better than the optimal solution set, this situation may occur when the optimization problem is strictly constrained. Including these solutions in the dataset increases the accuracy of the circuit model. When the solution is outside the first region of the objective space, it was not appended to the training data because these solutions contain ”failed” measurements that can not be expressed mathematically. Finally, in the last infeasible solution case where created offspring both have a worse rank than pareto optimal solution set and it does not satisfy predefined constraints, the decision to include them in the dataset is made by the user. Especially, if some solutions cluster at extreme points of the objective space, these solutions are not included in the dataset.

5.2. Training of Active Loaded Differential Amplifier

As the first example of ML-based optimization, the differential amplifier whose schematic provided in **Figure 5.3** has been synthesized. As the circuit works under a single supply, the gate terminals of the input pair are externally biased by a voltage source. The design parameters and objectives which also introduce to input layer and output layer of the ANN model, respectively, are listed in **Table 5.1**. The synthesized circuit was not constrained to test the circuit modeling and ML-based optimization abilities of the developed tool.

Table 5.1: Design parameters and objectives for the differential amplifier.

	L[μm]	W[μm]	R[Ω]	V_{bias}[V]	Objectives: M=2	
Lower Bound	0.12	0.16	100	0.2	Gain	Bandwidth
Upper Bound	1.3	97.5	10k	0.8	Max.	Max.

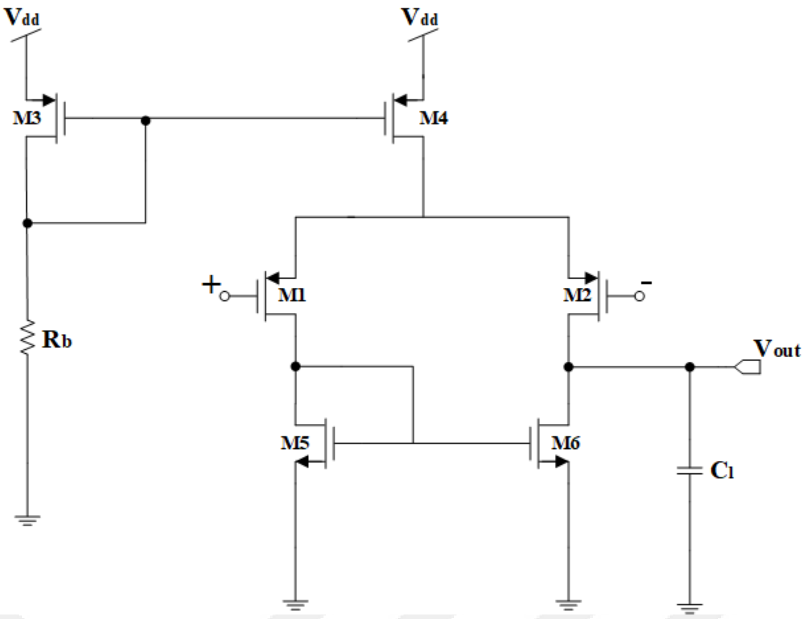


Figure 5.3: Active-loaded differential amplifier schematic.

A five-layer ANN model was constructed to mimic the active-loaded differential amplifier circuit behavior. The representation of the ANN model for the differential amplifier is shown in **Figure 5.4**. The input and output layers contain 8 neurons and 2 neurons, which are the number of all design parameters and design objectives, respectively. Three hidden layers are placed between the input and output layers, where all hidden layers are activated by rectified linear unit (ReLU) function and it contains 160, 300, and 300 neurons, respectively. The model has been trained with 150 epochs.

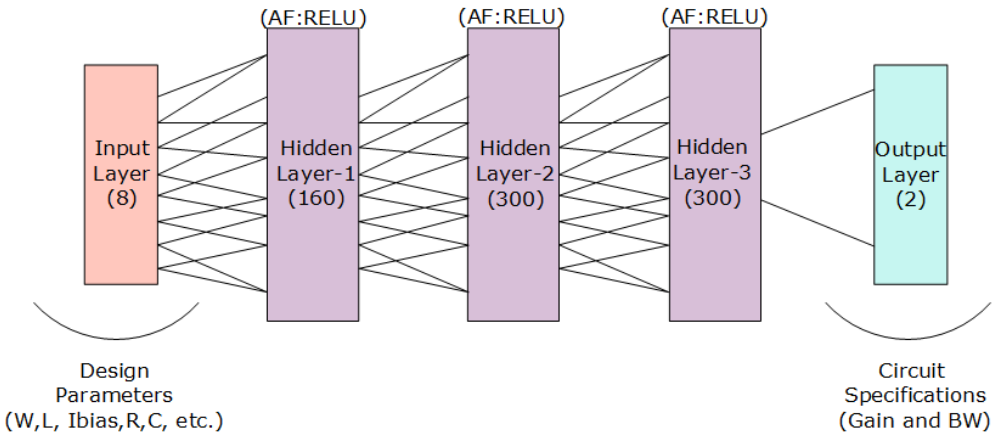


Figure 5.4: ANN model for active-loaded differential amplifier

The synthesis has been carried out with 100 generations for 100 population size. The model training has lasted up to 50 generations, and the circuit model has been utilized as the performance estimator instead of SPICE in the remaining 50 iterations. Thus the optimization process can be defined as SPICE-model combination-based synthesis. The obtained three PoFs are shown in **Figure 5.5**, where blue, green, and red indicators represent obtained PoF via only simulation-based synthesis, simulation and model combination-based synthesis, and only model-based optimization, respectively. As seen from the figure, the developed model has failed for only extreme solutions in objective space. This error might caused by the lack of adequate amount training data for extreme solutions. Nevertheless, model-based synthesis has been achieved to obtain true PoF.

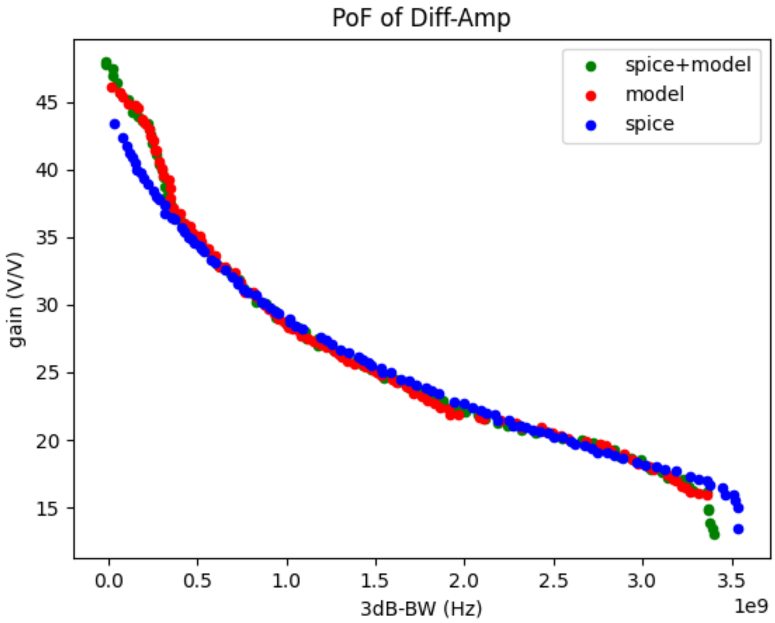


Figure 5.5: The constructed model performance for differential amplifier synthesis.

5.3. Training of Two-Stage OTA

The two-stage OTA whose schematic is depicted in **Figure 5.6** was selected as the second test circuit. As the circuit is symmetrically supplied, there is no need for gate biasing. In this context, transistor lengths and widths, compensation capacitor, and biasing resistor are the decision functions. As the amplifiers are usually utilized in closed-loop configuration in system-level design, a close-loop test bench has been constructed to make proper measurements for two-stage OTA. DC gain and unity gain frequency were determined as design objectives like in the previous synthesis. Additionally, to increase the complexity, phase margin, power consumption, and chip area have been defined as design constraints. All optimization parameters of the two-stage OTA are tabulated in**Table 5.2**.

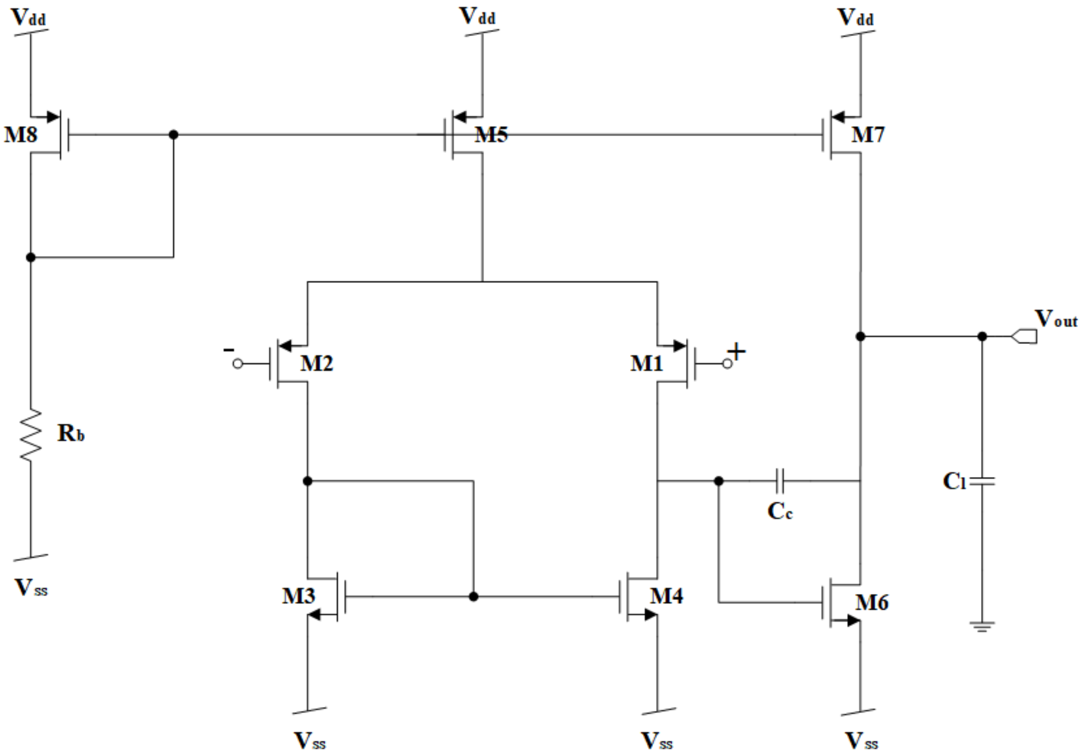


Figure 5.6: Two-stage OTA schematic.

Table 5.2: Design parameters and objectives for two-stage OTA.

Bounds	L[μm]	W[μm]	R[Ω]	C_C[pF]	Objectives: M=2		Cosntraints: K=3		
Lower	0.12	1	100	0.1	Gain	Bandwidth	PM	Power	Area
Upper	1.3	100	10k	50	Max.	Max.	>45°	<10mW	<500um^2

To increase the efficiency of the two-stage OTA modeling, different ANN models have been created for the modeling of each design constraint. Design objectives have been estimated with the help of a single ANN model, as in the previous example. Shortly, the constructed model for two-stage OTA was separated into sub-models which is shown in **Figure 5.7**. The training dataset which includes normalized design parameters and normalized performance values of the circuit has been directly used in the objective space and phase margin models' training. However, a classifying approach has been considered in constructing power and area estimators. All power and area constraint values were divided into two groups: Fall takes 0 and pass takes 1. The power and area models estimate to what percentage the constraint is satisfied rather than a direct estimate. In the ml-based optimization process, when the power or area model gives a value below 0.5, the circuit is considered to violate the defined constraint.

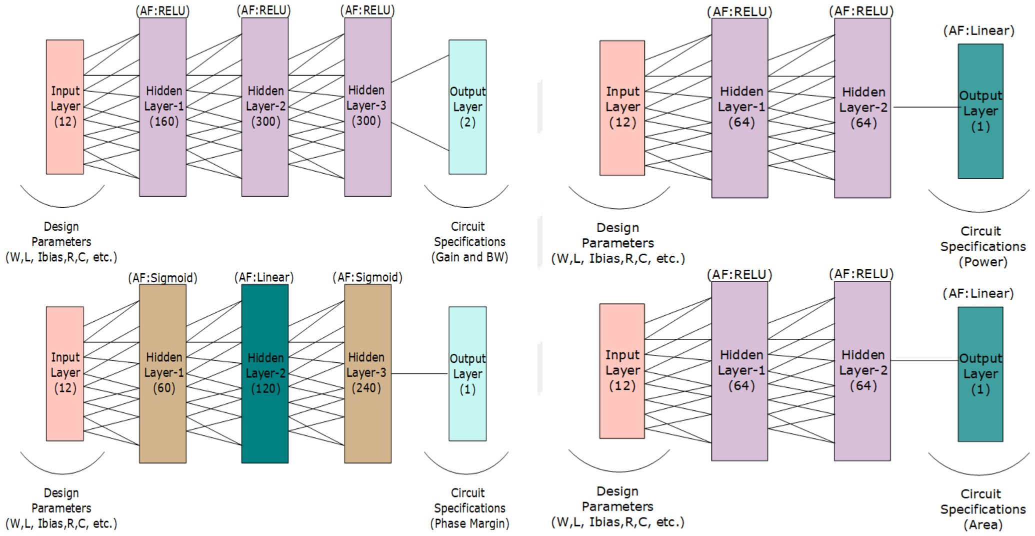


Figure 5.7: ANN model for two-stage OTA.

The network structure of design objectives is exactly the same as the differential amplifier network structure, except for the number of input layer neurons. The design objectives model has been trained with 400 epochs. On the other hand, the ANN model created for phase margin estimation contains three hidden layers with 60-120-240 neurons and sigmoid-linear-sigmoid activation functions, respectively, and the model was trained with 200 epochs. Two identical ANN models, which include two hidden layers with 64 neurons were developed to perform power and area estimations.

The two-stage amplifier circuit has been optimized for 300 generations with 150 individuals. The model has been trained with generated individuals by searching algorithm over the 200 generations and the synthesis process has turned into model-based optimization after 200th generation. The obtained PoFs via three different synthesis processes, where only model-based optimization fails in extreme solutions in the high gain corner are shown in **Figure 5.8**.

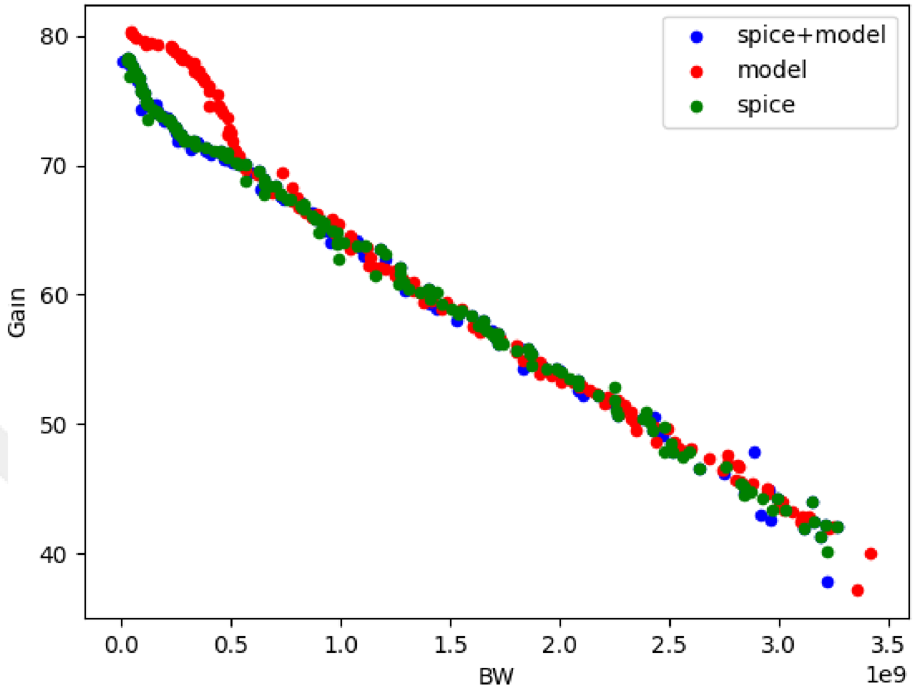


Figure 5.8: The comparison of performance evaluators for two-stage OTA synthesis.

5.4. Training of Voltage Comparator

The voltage comparator circuit whose schematic is shown in **Figure 5.9** is selected as the third benchmark circuit to test the developed model-based tool. The circuit drives a 0.5 pF capacitive load, and it is supplied by a 0.9V symmetric supply. Also, the current source of the active-loaded amplifier (M4) is biased by fixed voltage of 0.1V. While the transistor width values (pairs have the same width) and the length value (all transistors have the same channel length) introduce decision space, the power consumption and delay are both design objectives and constraints. All optimization parameters of the voltage comparator are listed in **Table 5.3**.

Table 5.3: Design parameters and objectives for voltage comparator.

Bounds	L[um]	W[um]	Objectives: M=2		Constraints: K=2	
Lower	0.13	0.16	Power	Delay	Power	Delay
Upper	1.3	80	Min.	Min.	<0.5mW	>0s

Two different ANN models have been constructed for the design objectives of the voltage comparator. Synthesis of the comparator circuit is a difficult process and many ”failed” measurements might be obtained during the synthesis. These ”failed” measurements were not added to the training data to ensure model accuracy.

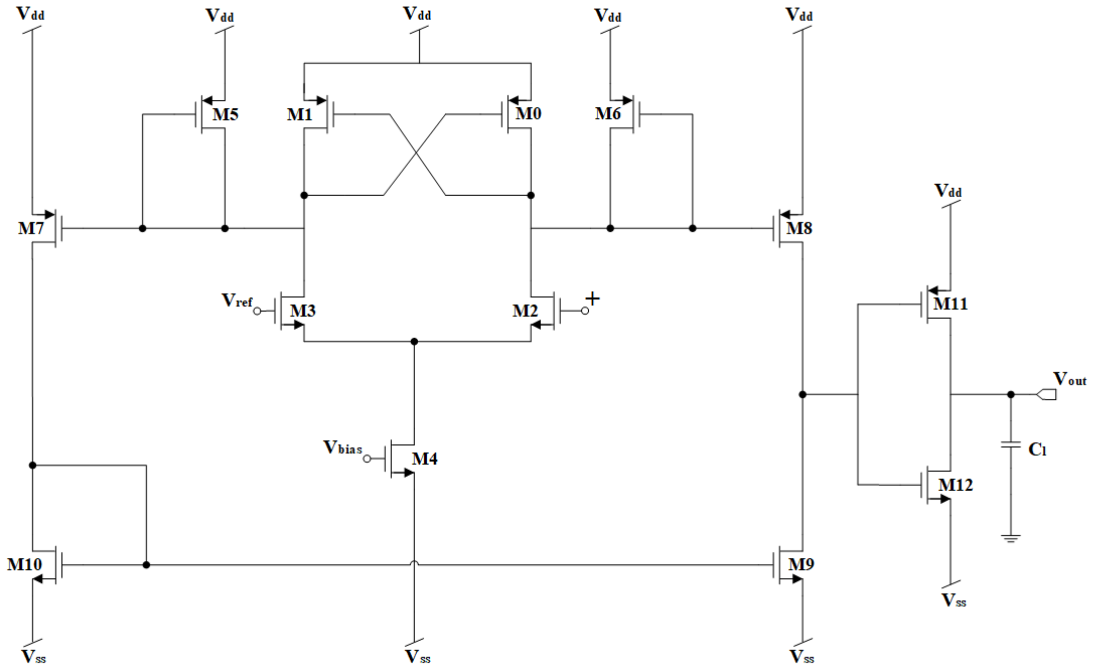


Figure 5.9: The schematic of the voltage comparator.

Two ANN models constructed to separately estimate the design objectives of the comparator are given in **Figure 5.10**. Both models contain an input layer, where neuron size is the number of design parameters in the decision space, three hidden layers, and an output model. The ANN model to perform power consumption estimation contains 160,300, and 300 neurons in the hidden layers all of them activated by RELU function, respectively. On the other side, the delay model contains three hidden layers with 20,30, and 40 neurons, where the exponential, exponential linear unit (ELU), and RELU functions were determined as activation functions, respectively.

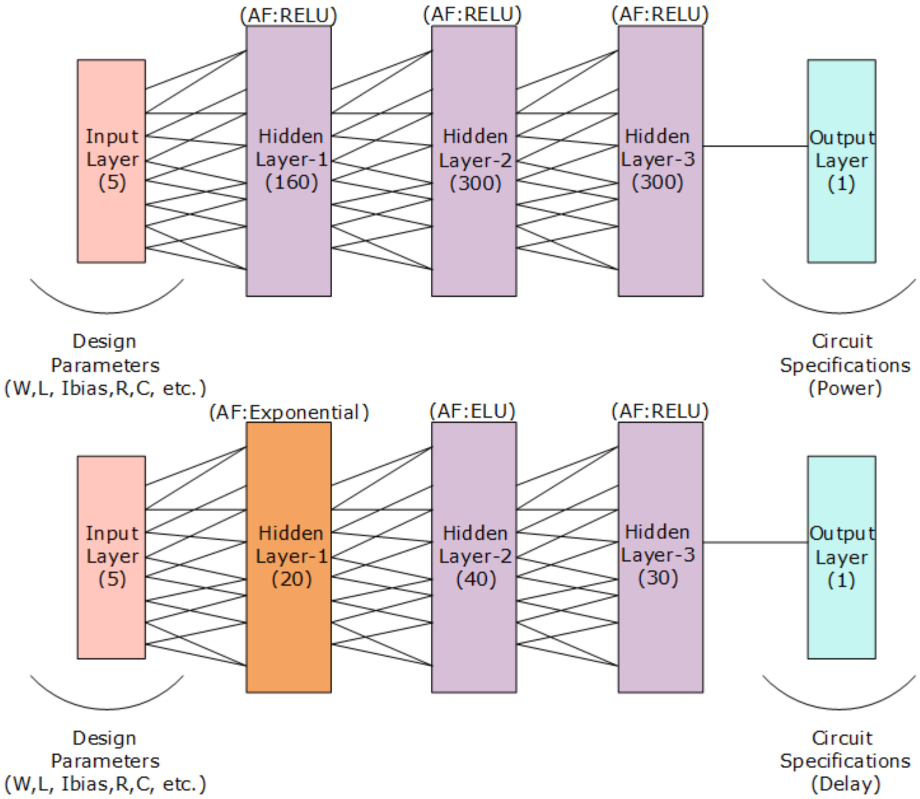


Figure 5.10: ANN models for voltage comparator.

The synthesis of the comparator has been carried out for 150 generations with 100 population size. The model training was completed in the first 90 iterations, and optimization continued using model evaluations until the maximum number of iterations was reached. The resultant PoFs of three different optimization approaches are given in **Figure 5.11**. It can be said that the voltage comparator was modeled accurately because both PoFs of model-based optimizations cover the PoF obtained by simulated-based optimization.

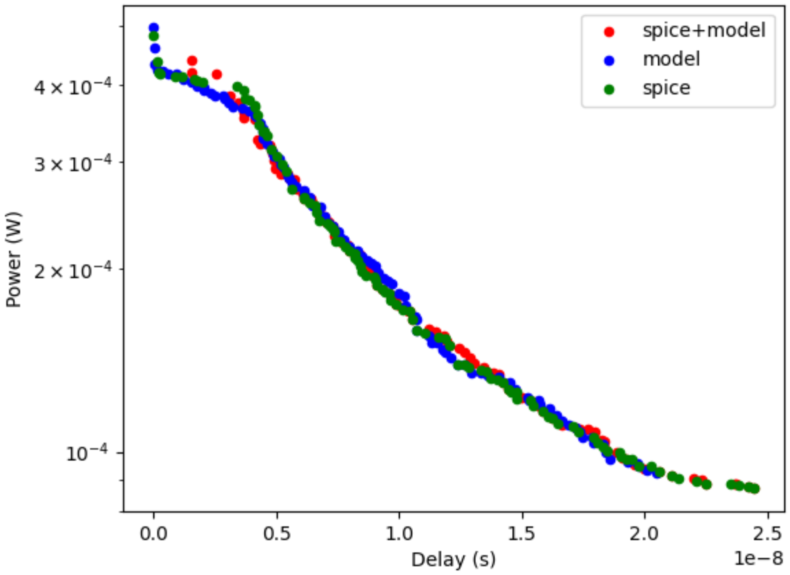


Figure 5.11: The constructed model performance for voltage comparator synthesis.

5.5. Training of Low-Noise Amplifier

An RF circuit, cascode LNA has been considered as the fourth benchmark circuit. The schematic of the circuit is shown in **Figure 5.12**. The design parameters which are transistor lengths (L_{1-3}), transistor widths (W_{1-3}), biasing resistors (R_1 and R_2), and inductor '2-π' model parameters (D_o,W_{ind},N_t, and s) are tabulated in **Table 5.4**.

Table 5.4: Decision space of cascode LNA circuit.

Boundaries	L[μm]	W[μm]	$R_1[\Omega]$	$R_2[\Omega]$	$D_o[\mu$m]	$W_{ind}[\mu$m]	N_t	s[μm]
Lower	0.12	24	1k	1k	75	2	2.5	1.5
Upper	1	480	15k	15k	150	10	7.5	2.5

The LNA circuit has been optimized to achieve the best solution by establishing a balance between power consumption and noise figure specifications that were determined as design objectives. On the other hand, other important design specifications for LNA such as S-parameters and power consumption were defined as design constraints. Determined design objectives and constraints of LNA optimization are listed in **Table 5.5**.

Table 5.5: Design objectives and design constraints of LNA circuit.

Objectives: M=2		Constraints: K=4			
Power	Noise Figure	Power	S_{11}[dB]	S_{12}[dB]	S_{21}[dB]
Min.	Min.	<10mW	<-15	<-30	>10

In the data set creating process, not all individuals created by the searching algorithm were included in training data. For the sake of ensuring accuracy, the failed solutions have been removed from the dataset, and the selected data interval from the whole objective space has been narrowed. Three ANN models have been constructed to mimic LNA circuit behavior. The power consumption and noise figure which are objective space functions are modeled with one ANN which has three hidden layers. Also, the precise estimation of S_{11} value another ANN model was created with five layers. The S_{12} and S_{21} parameters have not been directly estimated, these constraints have been trained via the classifying method, like in power and area models for two-stage OTA.

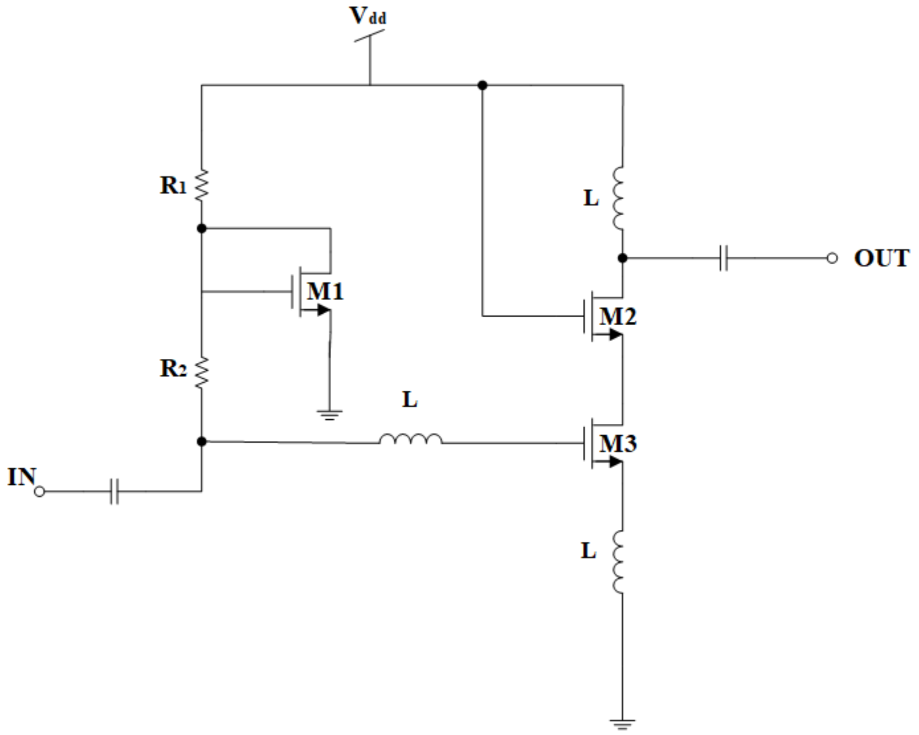


Figure 5.12: Schematic of the cascode LNA circuit.

ANN model to perform design objective prediction consisting of three hidden layers all of which include 50 neurons and activated by RELU function. The S_{11} value is estimated by the ANN model with three hidden layers, where the neuron sizes are 160, 300, and 300 respectively. For constraints S_{12} and S_{21}, an ANN consisting of two hidden layers for 64 neurons with activation function RELU, and a linear function output layer was modeled. The constructed ANN models are shown in **Figure 5.13**.

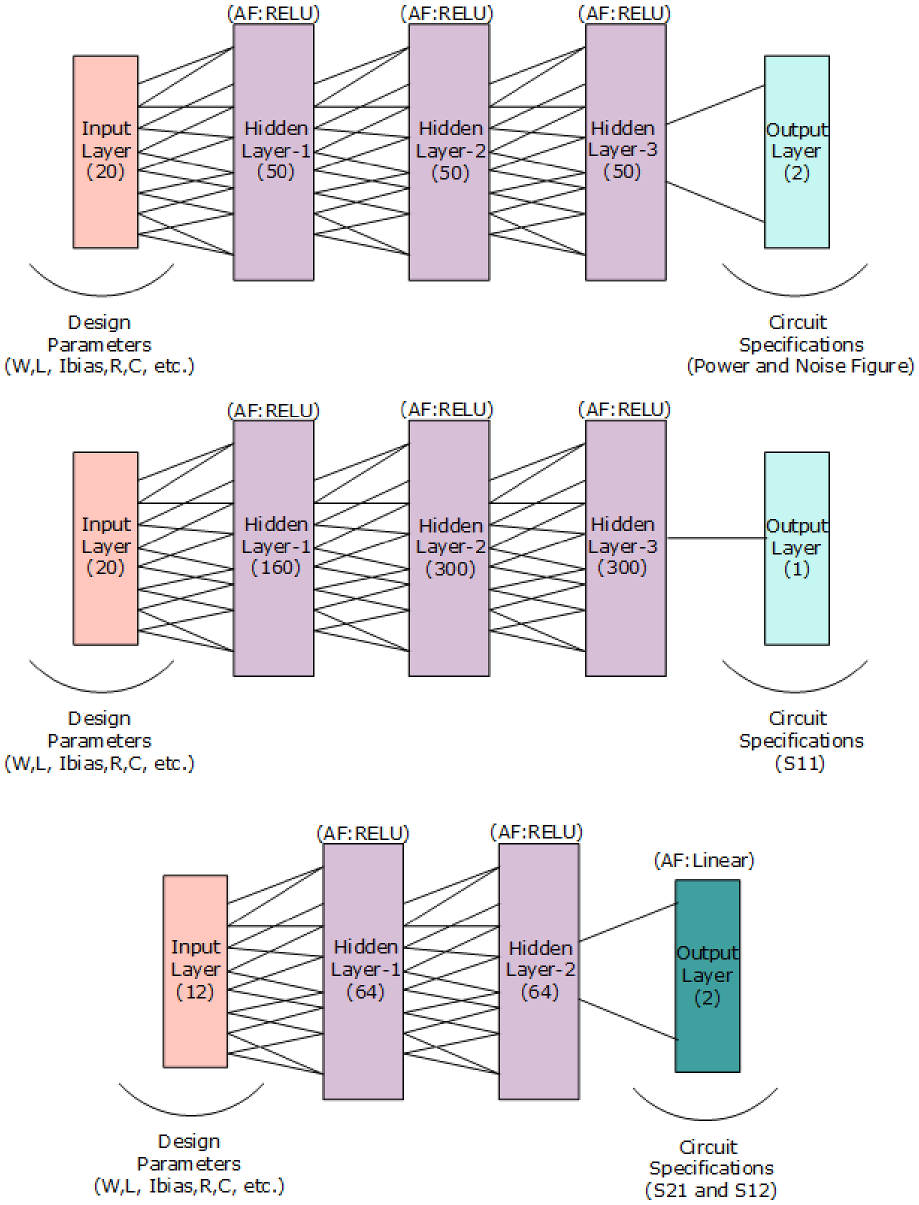


Figure 5.13: ANN models for low-noise amplifier

The circuit has been synthesized for 400 generations with 50 population size. Half of the synthesis was carried out by spice-based optimization, and the remaining half was continued with model-based optimization. As seen from the resultant PoFs given in **Figure 5.14**, obtained PoFs with model-based optimizations cover the real PoF.

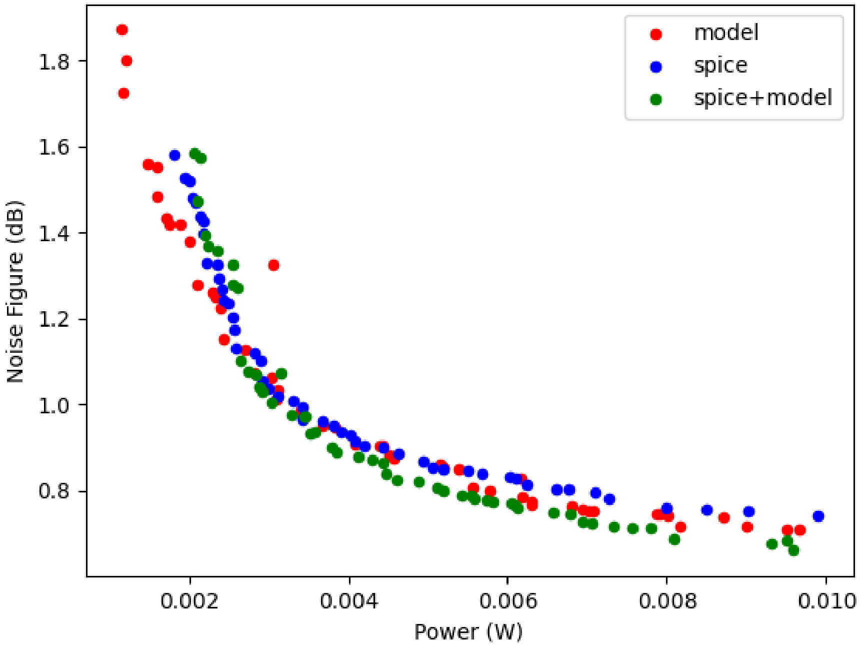


Figure 5.14: The comparison of performance evaluators for LNA synthesis.

5.6. Evaluation of Model Performances

Even though the effectiveness of the model was measured by performance metrics in the training phase, the robustness of the model can not be evaluated. There are numerous machine learning performance metrics, each is used to evaluate model performance from various aspects. In a training process, the whole dataset is not used for training, a certain percentage is reserved for testing, and performance evaluation is carried out using test data. The metrics results give important insights about the model performance, however, they suffer from limited test data drawbacks. For this reason, the real effectiveness of the created model can be seen when the training process is completed and used in the real world. In this context, the performance of the constructed circuit models has been evaluated through their pareto front approximations quality. The quality of obtained PoFs by ML-based optimization has been measured quantitatively via PoF metrics which are GD, IGD, SP, Δ, MS, and hypervolume (HV). The hypervolume indicator gives the volume of the space between pareto front approximation and a reference point dominated by each solution in pareto optimal set. An illustration of the HV indicator is shown in **Figure 5.15**.

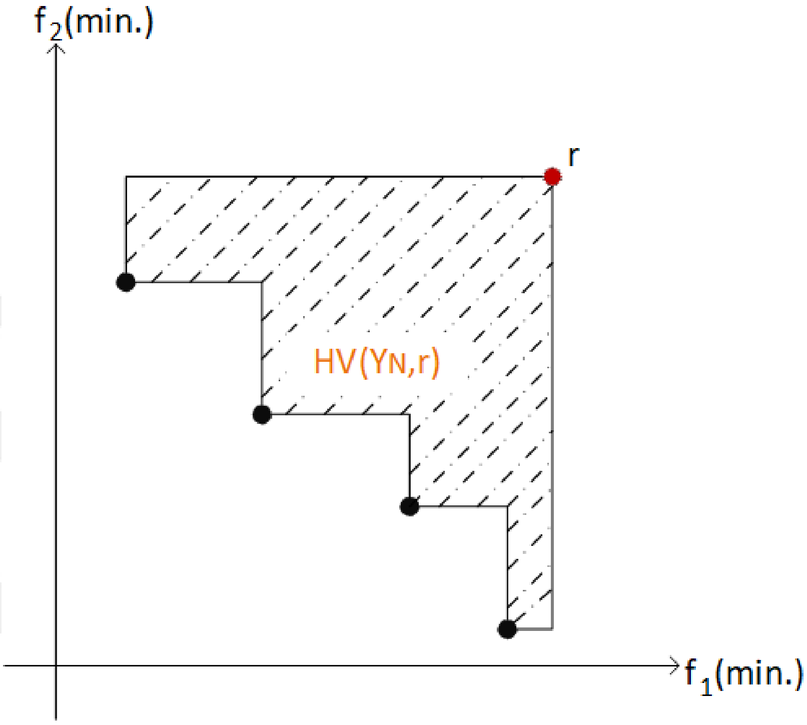


Figure 5.15: Illustration of hypervolume indicator for multi-objective problem.

Also, in the case presence of true pareto optimal front, the Hyperarea (HR) ratio given in Equation 5.1 is used to determine the convergence of obtained PoF. The greater HR value means that better convergence is achieved. However, the HR metric may be misleading regarding the convergence information in our case, as the ANN model may evaluate the circuit performance a little bit optimistically. For instance, if the obtained PoF is more converged than the true PoF, it is concluded that the constructed model is not performing properly, even if the HR ratio goes above 1. To address this issue, a basic approach named derived hyperarea ratio (DHR), inspired by HR, is proposed in Equation 5.2.

$$HR(Y_N, Y_P; r) = \frac{HV(Y_N; r)}{HV(Y_P; r)} \tag{5.1}$$

$$DHR(Y_N, Y_P; r) = \left(1 - \frac{|HV(Y_P; r) - HV(Y_N; r)|}{HV(Y_P; r)}\right) \tag{5.2}$$

All of the calculated quality metrics for benchmark examples PoFs (model-based optimization) are listed in **Table 5.6**. As the reference PoF is needed to calculate metrics except spacing, PoF obtained via simulation-based optimization was considered true PoF. As seen from the metric results, all of the benchmark circuits have been successfully modeled. Also, DHR is utilized to determine the accuracy of the circuit model. According to the DHR results, active-loaded differential amplifier, two-stage OTA, voltage comparator, and low-noise amplifier have been modeled with 99.5%, 95%, 99.7%, and 95.1% accuracy, respectively.

Table 5.6: Summarizing of model-based synthesis results with quality scores.

	GD[↓]	IGD[↓]	SP[↓]	Δ[↓]	MS[↑]	DHR[↑]
Differential Amplifier	1.85e-3	1.88e-3	4.37e-3	0.3897	0.9335	0.995
Two-stage OTA	2.17e-3	2.08e-3	7.77e-3	0.6897	0.9971	0.95
Voltage Comparator	1.35e-3	3.21e-3	8.72e-3	0.4324	0.9152	0.997
Low-Noise Amplifier	4.35e-3	2.27e-3	14.34e-3	0.7283	0.9857	0.951

6. HIERARCHICAL SYNTHESIS EXAMPLES

As it is known, most consumer electronics products today require system-level design of integrated circuits, which is the combination of analog or digital sub-blocks. The increase in demand for consumer electronics has brought the need to design more complex systems in a shorter time. At this point, hierarchical design automation emerges as a method that can overcome the time and complexity bottleneck. Two basic approaches are generally used to perform the hierarchical design of an analog/RF system: Bottom-up and top-down design approaches. The designer chooses which method to use according to the system requirements.

The bottom-up hierarchical design process was developed by dividing a system into sub-blocks that can be easily designed and make sense when combined. First, The sub-blocks that make up the parts of the system are determined. To meet the performance criterias of the system, certain design constraints are determined for each sub-block circuit, and appropriate circuit topologies are selected according to these constraints. Circuits designed at the lower level are combined to obtain blocks at the upper level. The new block created at the top level will be the building block of the next level. This process continues until it reaches the system level. The biggest difficulty of this method, which is generally preferred by designers, is the need to reconstruct the design at the lower level if even one of the design requirements at the upper level is not met. The designer may need many iterations to complete the system-level (top-level) design. When evaluated from this perspective, the bottom-up hierarchical design flow is a challenging and quite time-consuming process.

On the other hand, the top-down hierarchical design method can also be used to design a system. To reduce the time and effort in the bottom-up hierarchical design method, designers use circuit/block models at the upper levels. As can be seen from the figure, each circuit model is reevaluated at lower levels. This re-evaluation process is essential in the top-down design process because behavioral languages such as Verilog-A/AMS, VHDL-AMS, or System-Verilog, or simulation environments such as Matlab, used to model the upper levels, are difficult to model highly nonlinear circuits. Creating a high-accuracy behavioral model at the system level can be also very time-consuming and requires serious effort.

In both hierarchical design methods, it is not possible to scan the entire possible design space by exerting manual effort. On the other hand, synthesizing the system-level design with the help of an optimization tool is far from practical. In this context, PoFs created for the lowest level can be used to perform space scanning at the system level. PoFs obtained by simulation-based optimization for lower-level circuits are combined through upper-level mathematical expressions. Thus, for an analog/RF system, space exploration occurs within seconds. To test the developed method, two different equation-based hierarchical synthesis examples are presented in this chapter.

6.1. Hierarchical Synthesis of Multi-Stage Wide-Band Amplifier

The authors in [68] have proposed a hierarchical synthesis approach for cascaded wide-band amplifiers. The proposed approach is shown in **Figure 6.1** which also can be employed for three or more stage amplifiers. Firstly, a single-stage fully differential amplifier (SSFDA) has been optimized via a MOO tool to obtain PoF to utilize hierarchical synthesis. Once obtained PoF of SSFDA has been used to construct the cascaded two-stage amplifier through hierarchical synthesis methodology. To obtain PoF for the two-stage amplifier, the cross product of PoFs was performed, in which individuals selected from one front were combined with all points from the other front. The process was continued until there were no individuals left to perform the combination in the first PoF. To construct PoF for the two-stage amplifier, only the non-dominated solutions were selected from all candidates which formed from the result as a hierarchical combination. The same procedure was applied to perform three-stage amplifier synthesis, where the first and second stages were considered as a single amplifier. All combination operations performed in hierarchical synthesis are supported by mathematical equations. More information about this method can be found in [68].

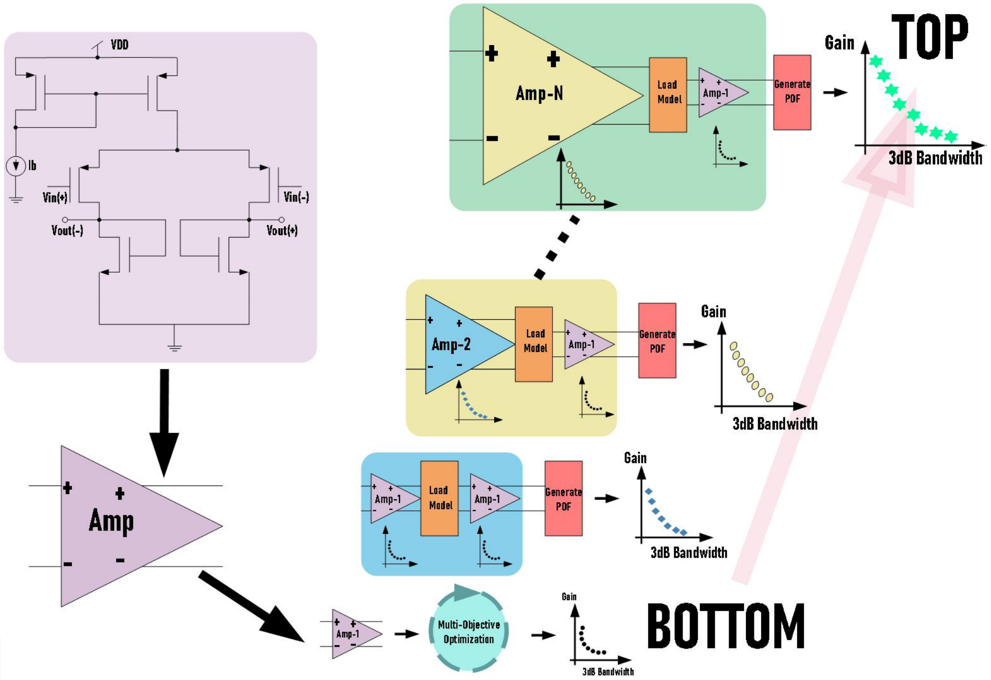


Figure 6.1: General framework of the hierarchical approach proposed in [68].

The intermediate frequency amplifier (IFA) which is a building block of super-heterodyne receiver circuit has been considered the first hierarchical synthesis example. The IFA amplifier is formed by three cascaded amplifiers. To achieve the best possible solutions in the design space of IFA, a single-stage fully differential amplifier whose schematic is given in **Figure 6.2** has been synthesized with hierarchical synthesis methodology. Although the topology seems similar to the active-loaded differential amplifier topology, the output resistance is quite low compared to the output resistance of the active-loaded amplifier as both load transistors are in the diode connection. Low output impedance offers high bandwidth at the expense of gain. Since the circuit operates with a single supply voltage, the input transistors were externally biased to operate in the appropriate region. The power constraint has been introduced to ensure that the power consumption of the circuit is lower than a certain level. Considering a multistage amplifier, the input transistors of each stage are biased with the DC voltage at the output nodes of the previous circuit. Therefore, a certain range was defined as a constraint on the DC voltage level of the circuit's output nodes. The design constraints, parameters, and objectives determined for the fully differential single-stage amplifier circuit are listed in **Table 6.1**.

Table 6.1: Design parameters and objectives for SSFDA.

Bounds	L[μm]	W[μm]	R[Ω]	V_b[V]	Objectives: M=2		Cosntraints: K=2	
Lower	0.13	0.65	100	0.5	Gain	Bandwidth	Gain	Power
Upper	1.3	100	15k	0.7	Max.	Max.	>3dB	<10mW

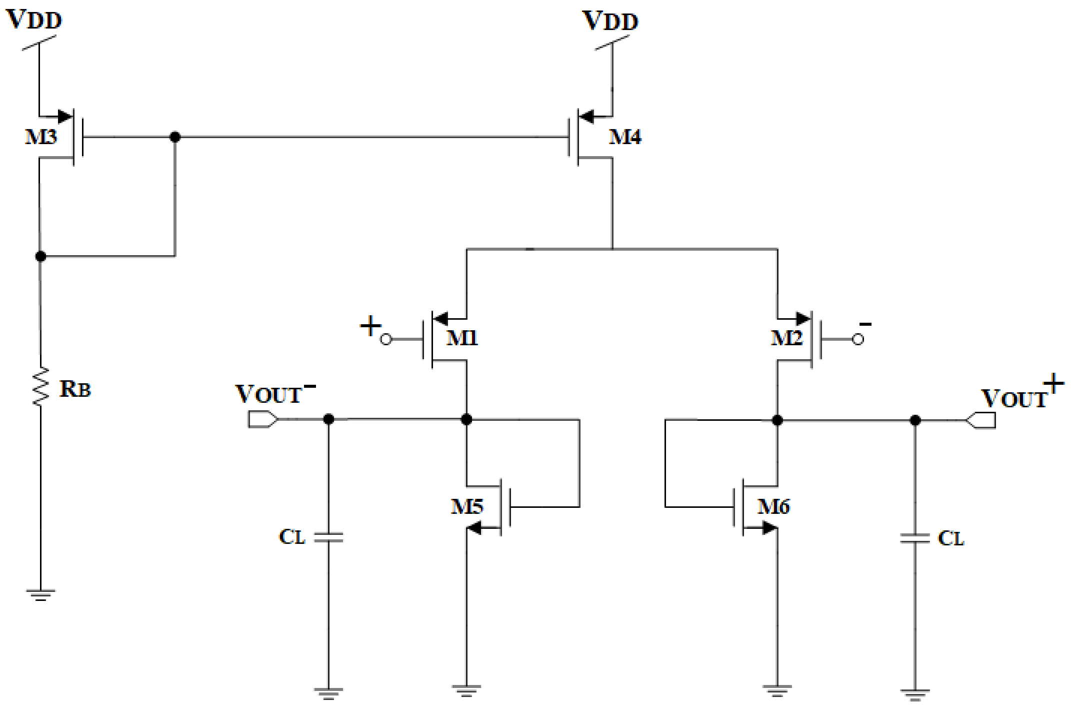


Figure 6.2: Schematic of single stage fully differential amplifier.

The single-stage circuit has been optimized for 200 generations with 100 individuals, where HSPICE was the performance estimator. The resultant PoF is shown in **Figure 6.3**. The synthesis result shows that the circuit is useful for applications that require high bandwidth. In addition, high gain and high bandwidth can be achieved at the same time with cascading of single-stage circuits.

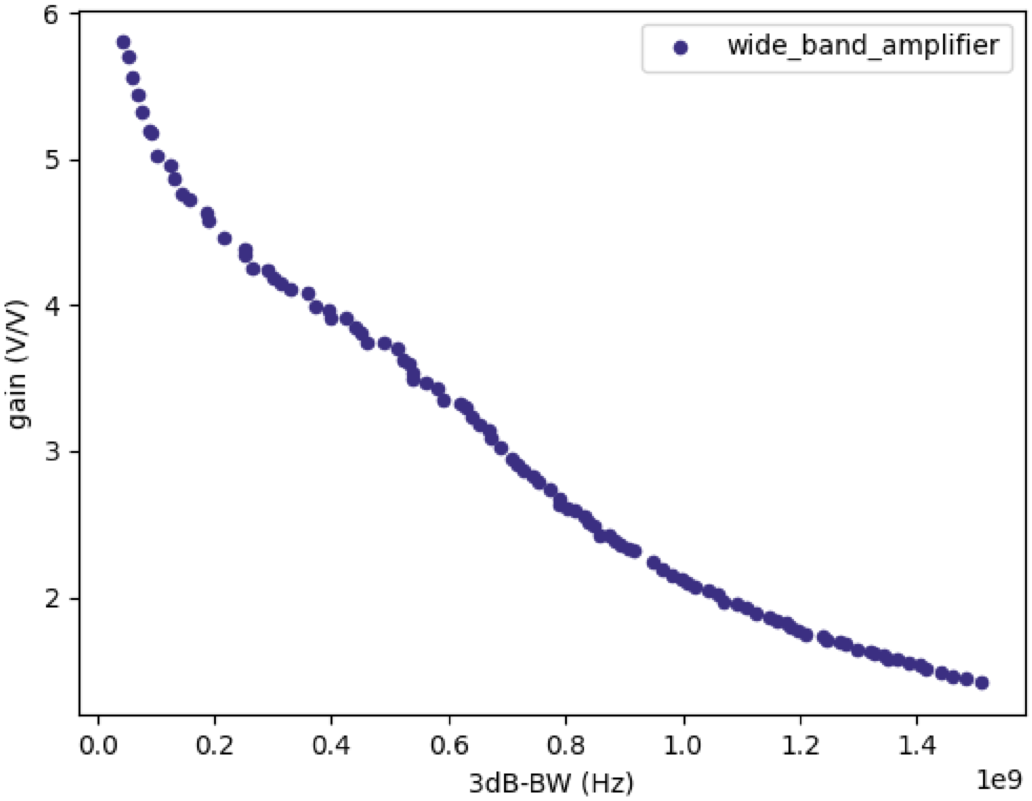


Figure 6.3: Obtained PoF of SSFDA.

The voltage gain and bandwidth can be easily calculated by Equation 6.1 and Equation 6.2, respectively, to estimate the performance of the cascaded amplifier. However, the bandwidth equation is not enough alone to model circuit behavior properly. Thus, the load effects of the previous stages must be taken into account.

$$A_{v,total}(V/V) = A_{v1}.A_{v2}.A_{v3}....A_{vn} \tag{6.1}$$

$$f_{3dB,total} = \frac{1}{\sqrt{\frac{1}{f_1^2+f_2^2+f_3^2...f_n^2}}} \tag{6.2}$$

To perform two-stage cascaded amplifier synthesis, the bandwidth of the first stage was estimated using the load model whereas, measured bandwidths in the PoF were directly utilized for the second stage because the load of the cascaded systems is the load capacitance of the synthesized single-stage circuit. Both self-capacitances of the first and second stages which are denoted by C_{int} and C_{ext}, respectively, are elements of the load model. The capacitance value at the output of the first stage can be estimated by Equation 6.3 and, Equation 6.4 which consist of transistor capacitances can be calculated by Equation 6.5, Equation 6.7, and Equation 6.6.

$$C_{int} = C_{db,n} + C_{db,p} + C_{gs,n} \tag{6.3}$$

$$C_{ext} = C_{gs,p} + (1 + A_v)C_{gd,n} \tag{6.4}$$

$$C_{gs} = \frac{2}{3}W.L.C_{ox} \tag{6.5}$$

$$C_{gd} = W.C_{gdo} \tag{6.6}$$

$$C_{db} = \frac{AD.CJ}{(1 + \frac{V_{DB}}{PB})^{MJ}} + \frac{PD.CJSW}{(1 + \frac{V_{DB}}{PBSW})^{MJSW}} \tag{6.7}$$

In the following step, output resistance was calculated from the PoF of the single-stage amplifier by Equation 6.8, where a known load (C_L) is used in the generation of the PoF. Finally, the new bandwidth value of the first stage was calculated using Equation 6.9.

$$R_o = \frac{1}{2\pi f C_L} \tag{6.8}$$

$$f = \frac{1}{2\pi R_o(C_{in} + C_{ext})} \tag{6.9}$$

Since a total of 100x100=10000 individuals were obtained by combining two PoFs, only non-dominated solutions were selected to obtain PoF for the two-stage cascaded amplifier. Also, obtained 10000 individuals through the hierarchical synthesis approach was simulated via HSPICE to observe the accuracy of the analytical modeling. The obtained PoFs for the cascaded two-stage amplifier are shown in **Figure 6.4**. It can be said that the constructed hierarchical model achieved true PoF in a few seconds.

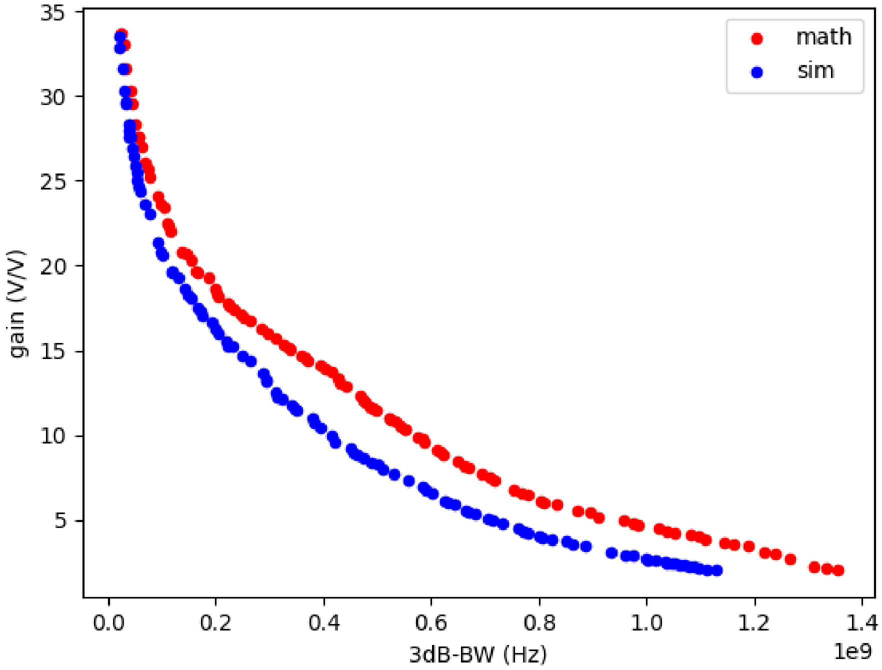


Figure 6.4: Obtained PoFs for two-stage cascaded amplifier.

To obtain the three-stage cascade amplifier, a single-stage amplifier in **Figure 6.2** was connected to the hierarchical model of the two-stage amplifier. An illustration of the three-stage cascade amplifier model is given in **Figure 6.5**, where the first and second stages are considered as a single block which PoF obtained previously. The flow used to estimate the PoF for the three-stage circuit is the same as the method applied for the hierarchical synthesis of the two-stage cascaded amplifier. It is worth mentioning that the occurred self capacitances of the second amplifier introduce internal capacitance (C_{int}). Input self-capacitances of the last stage constitute external capacitance (C_{ext}).

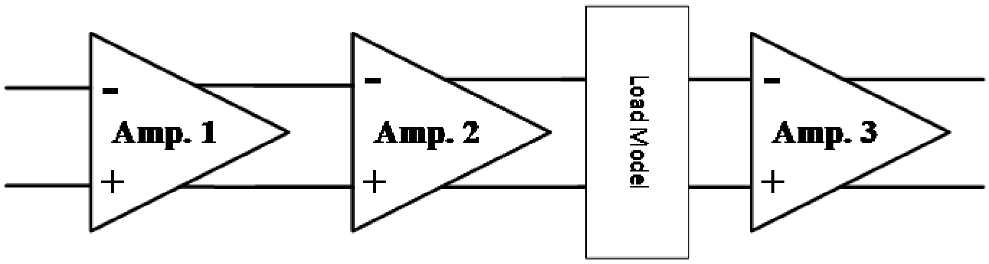


Figure 6.5: Load modeling representation for the three-stage amplifier.

The obtained PoF for the three-stage cascaded amplifier through the hierarchical synthesis method is given in **Figure 6.6**. The obtained PoF was compared with the PoF created with the simulation-based approach. The accuracy of the mathematical modeling-based hierarchical synthesis method has also been proven for the three-stage amplifier.

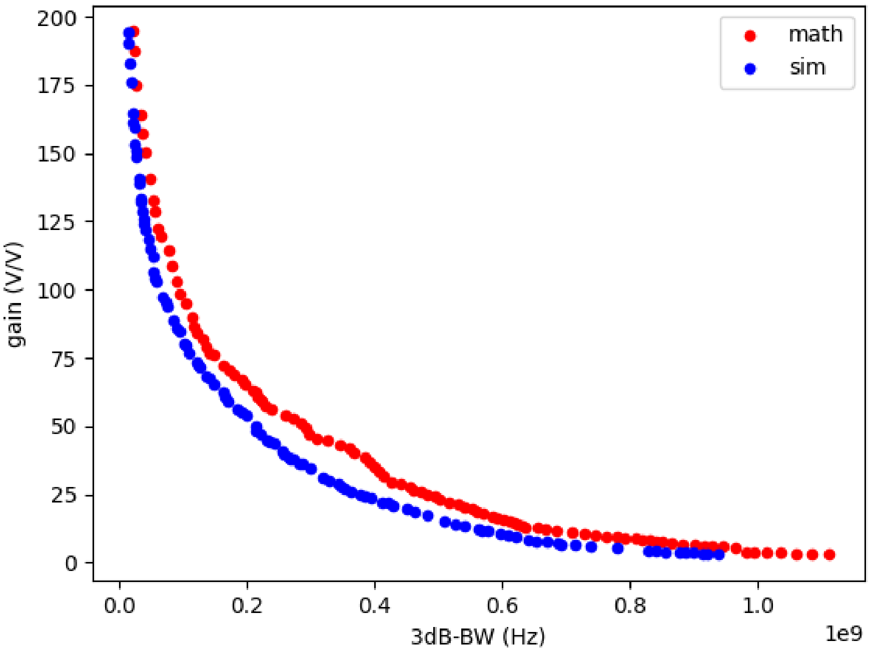
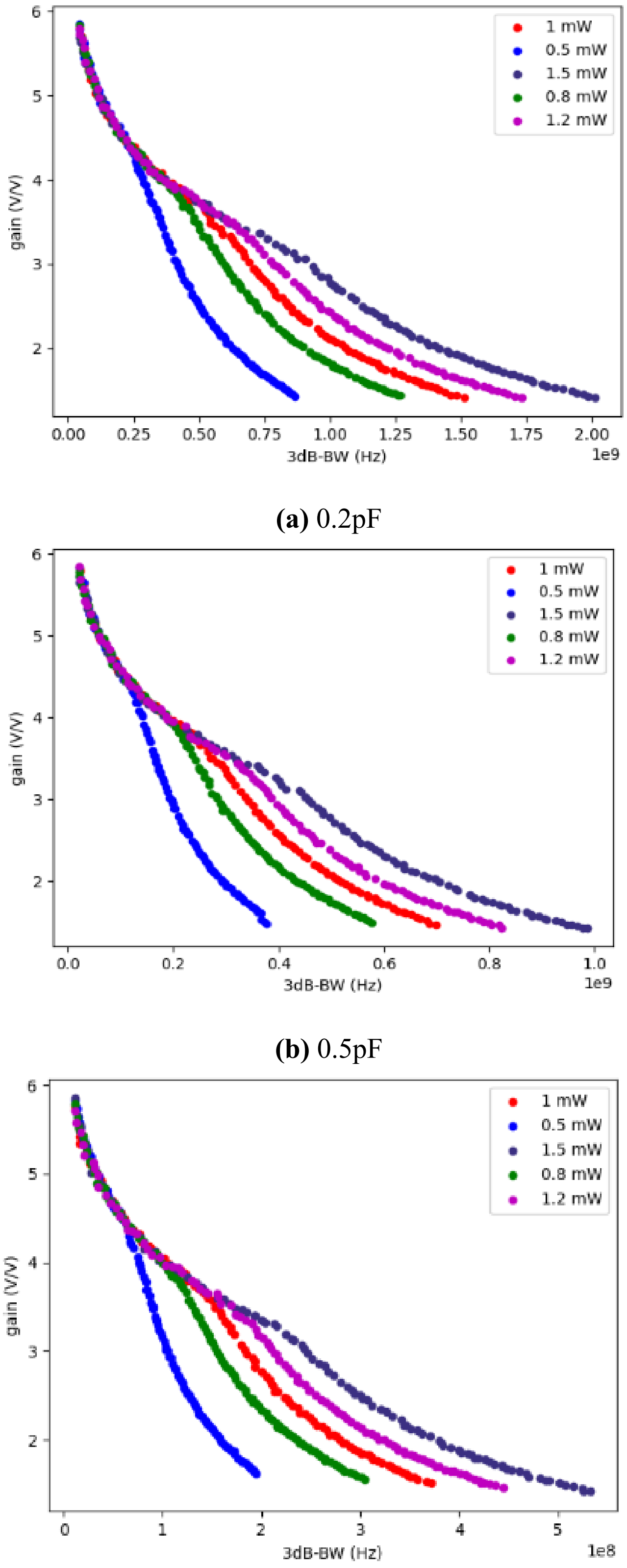


Figure 6.6: Obtained PoFs for three-stage cascaded amplifier.

On the other hand, how to do the design in given system-level constraints is another issue. To examine this situation, a certain power consumption limit has been introduced to two- and three-stage cascaded amplifier circuits driving different loads. In manual design, the designer divides the introduced power consumption limit among the stages of the amplifier according to the designer's own experience. However, iterations to test whether the total system reaches its best performance through manual design are quite tedious and impractical. In this context, hierarchical optimization ensures that the best possible solutions are found under the specified system constraint. Pareto-optimal fronts of a single-stage amplifier circuit driving different loads and having different power constraints are given in **Figure 6.7**.

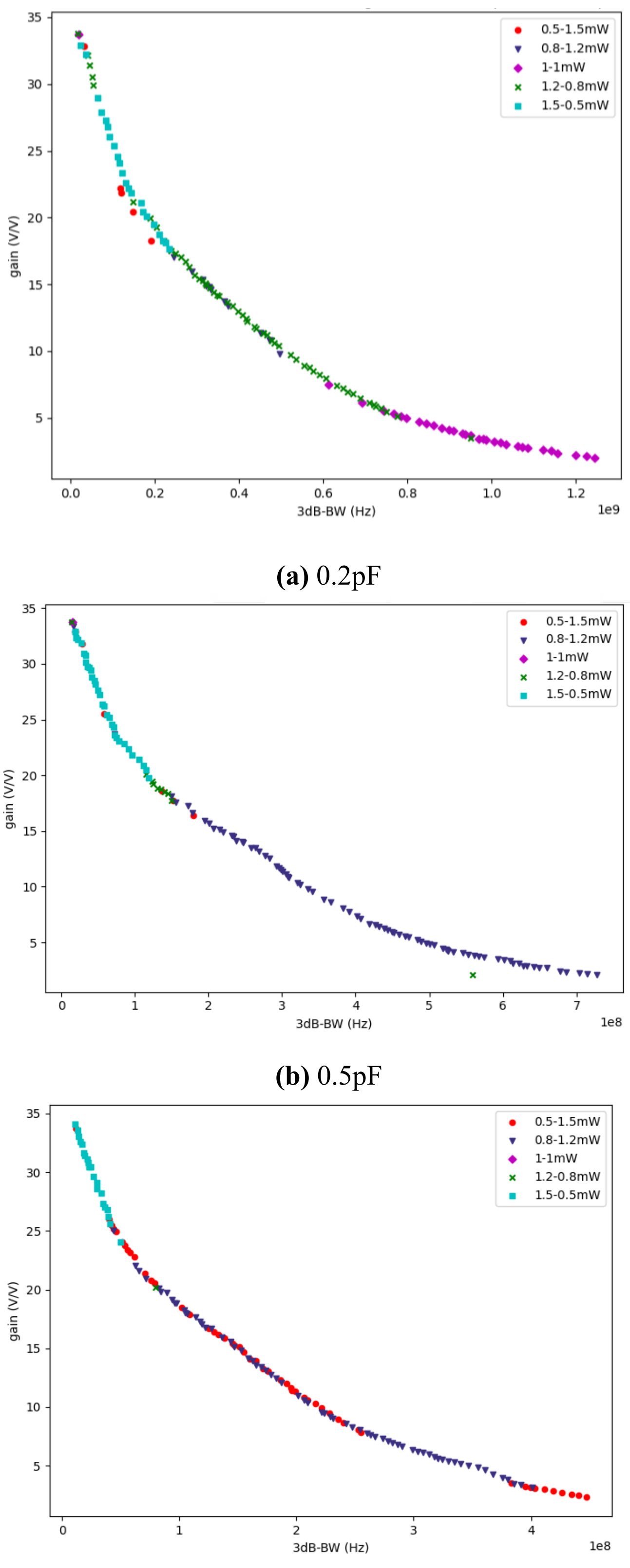


(a) 0.2pF

(b) 0.5pF

(c) 1pF

Figure 6.7: Obtained PoFs under different power constraints of the SSFDA.



(a) 0.2pF

(b) 0.5pF

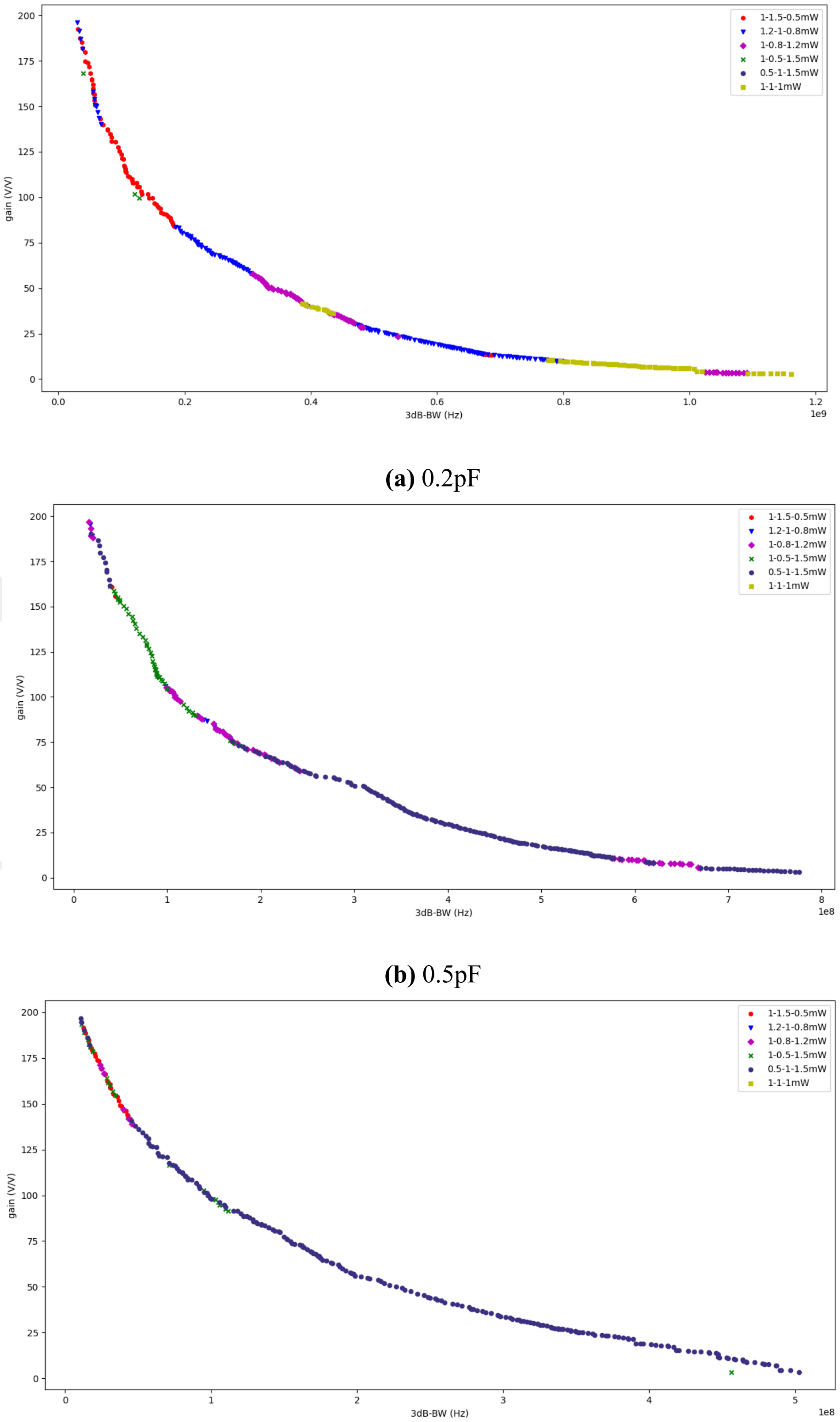
(c) 1pF

Figure 6.8: Obtained PoFs through hierarchical synthesis of cascaded amplifier.

After obtaining PoFs for different power constraints, the two-stage cascaded amplifier was combined hierarchically using the analytical model. The nuance here is that Pareto-optimal solutions of circuits with different power consumption constraints are combined. With a total power consumption constraint of 2mW, for example, individuals with a power consumption constraint of 0.5mW and individuals with a power consumption of 1.5mW were combined using the same methods as before, and the newly obtained individual set was named as the solutions obtained with the combination of 0.5-1.5mW. In this context, the effectiveness of power combinations for different loads was examined. The obtained PoFs via hierarchical synthesis approach are shown in **Figure 6.8**. To achieve high gain for all three cases, the power consumption of the second amplifier driving the load is low. On the other side, high power consumption of the second stage might be needed to obtain high bandwidth. As the load increases slightly, high gain can still be achieved with the 1.5mW-0.5mW combination, while high bandwidth can be achieved by the second stage consuming more power rather than the combination of identical circuits. For the case where the two-stage broadband amplifier drives a 0.5pF load, the combination of 0.8-1.2mW power consumption is quite dominant. In case the load driven by the circuit increased by 5 times (1pF), the circuits were selected which were constrained with 1.5mW.

Pareto optimal fronts were obtained with different power combinations for the three-stage cascaded amplifier. In this example, the total power consumption constraint was determined as 3mW. The obtained PoFs of the amplifier for different capacitive loads are given in **Figure 6.9**. It is easily seen that as the load increases, the power consumption of the last stage increases to achieve high bandwidth.

As a result, while designing at the system level, a high-accuracy model was used to combine the lower-level circuits, and the IFA block was designed by means of this model without any simulation. In addition, it has been shown that the obtained model works under varying power consumption and load capacity constraints, and sub-circuit design preferences have been examined according to the constraints given at the upper level.



(a) 0.2pF

(b) 0.5pF

(c) 1pF

Figure 6.9: Obtained PoFs through hierarchical synthesis for three-stage amplifier.

6.2. Hierarchical Synthesis of Voltage Comparator

The comparator circuit, which is the most important analog block of analog-to-digital converters (ADC) used in almost all circuits containing most electronic equipment, often requires a pre-amplifier at its input. It is a known fact that there is an improvement in delay performance when the voltage comparator circuit is used with a pre-amplifier. However, the pre-amplifier is generally designed with a gain of 5-10 (V/V). The comparator circuit was modeled mathematically at the systemic level to determine the relationship between the voltage gain of the amplifier and the speed of the comparator to prevent unnecessary resource usage. Pre-amplifier and comparator blocks have been optimized to perform hierarchical synthesis. The voltage comparator, which was presented in the previous sections as the first sub-block and whose scheme is given in **Figure 4.8**, has been synthesized using the simulation-based MOO tool. The transistor provides current to the input of the circuit is biased by 0.1V constant voltage. The transistor dimensions of the inverter, which forms the output stage of the circuit are determined by the designer, and they are not changed by the algorithm during the optimization process. Also, all transistor lengths have kept in same value, except for inverter transistors, to facilitate convergence. An amplitude of 0.2V PWL signal was applied to the positive input to evaluate the speed performance of the circuit. All design parameters of the voltage comparator are listed in **Table 6.2**.

Table 6.2: Design space of the voltage comparator.

W_{min} / W_{max}	0.16μm / 80μm
L_{min} / L_{max}	0.13μm / 1.3μm
V_{in} / V_{ref}	PWL (0n -0.1 50n 0.1) / 0V
V_{bias}	0.1V
V_{DD} / V_{SS}	+0.9V/ -0.9V
C_L	0.2pF
W_{11} / W_{12}	30μm / 8μm
L_{11} / L_{12}	0.13μm / 0.13μm

The circuit has been synthesized to achieve the best solutions in the trade-off between power consumption and time delay, which were also determined as design constraints. The obtained pareto solutions through MOO with different power constraints are scattered in the same plot. The resultant PoFs for the voltage comparator, when the load is 0.2pF, are shown in **Figure 6.10**.

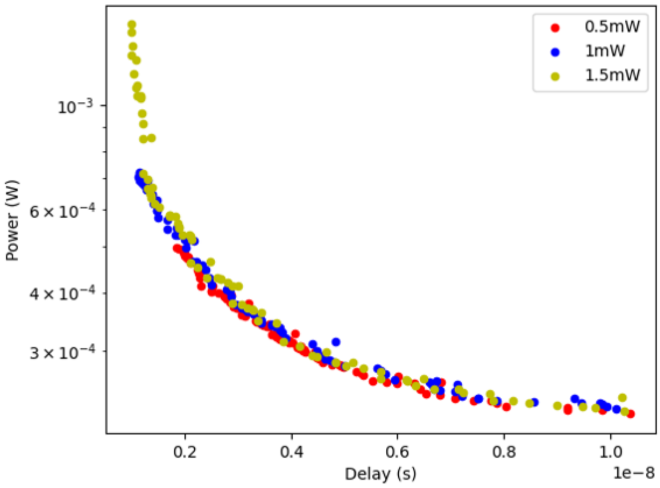


Figure 6.10: Obtained PoF for voltage comparator with different power constraints.

Table 6.3: Design parameters and constraints for resistive load amplifier.

Desicion Space		Design Constraints	
W_{min} / W_{max}	0.65μm / 157.5μm	Power	<0.5mW, 1mW, 1.5mW
L_{min} / L_{max}	0.13μm / 1.3μm	Gain	>3dB
$R_{b,min}$ / $R_{b,max}$	50/20kΩ	DC@Vout	0<Vout<50mV
$R_{L,min}$ / $R_{L,max}$	100/25kΩ		

A resistive load amplifier, whose schematic is shown in **Figure 6.11**, was chosen as the pre-amplifier to be used in front of the voltage comparator. The circuit is biased by a symmetrical supply voltage of 0.6V and it drives 0.2pF. The three optimizations have been carried out under different power consumption constraints. The optimization parameters of the pre-amplifier are listed in **Table 6.3**.

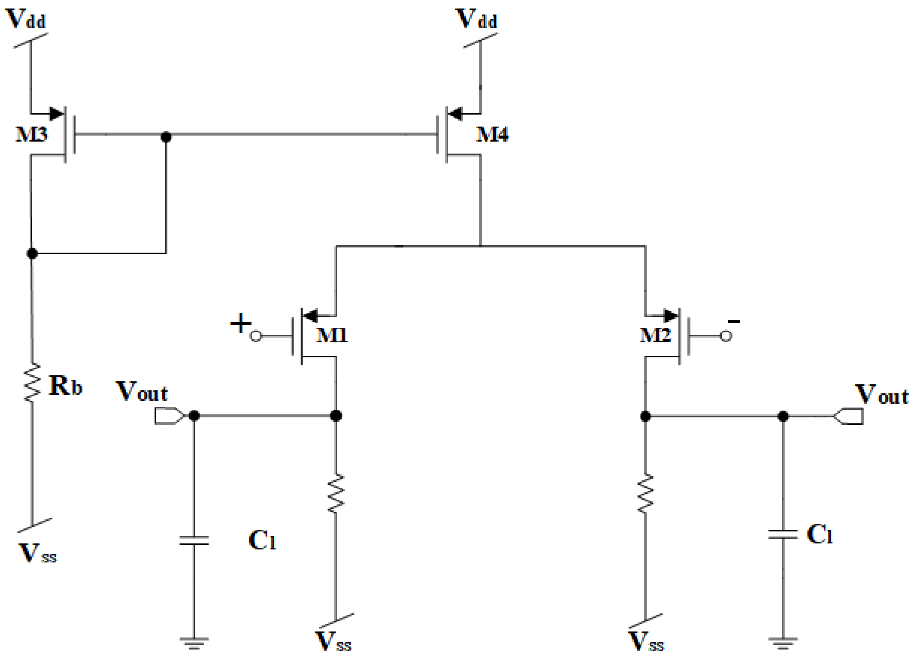


Figure 6.11: Schematic of the resistive load differential amplifier.

The circuit has been synthesized for 300 generation with 100 individuals. The resultant PoFs obtained through optimization for different power consumption is shown in **Figure 6.12**.

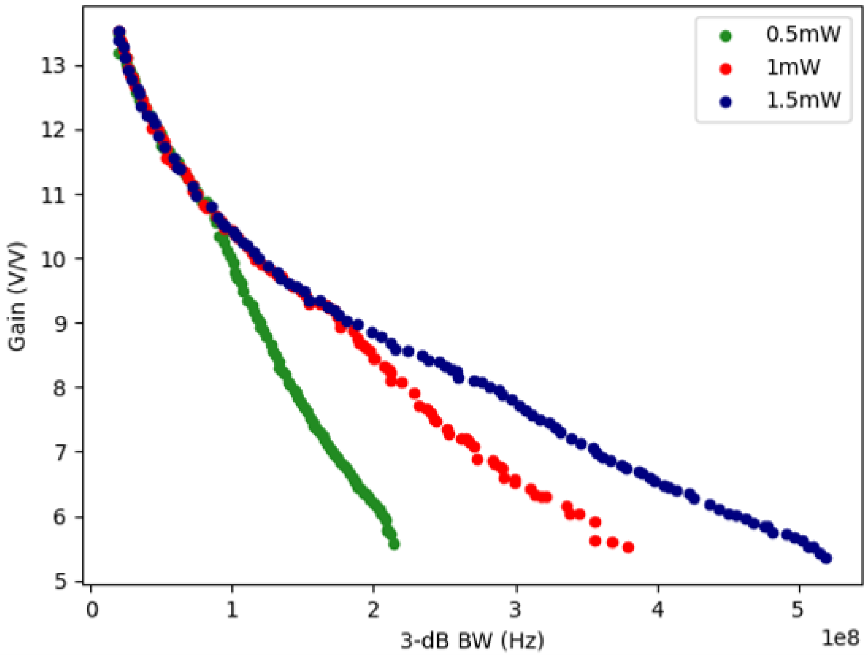


Figure 6.12: Resultant PoFs for resistive load amplifier optimization.

The pre-amplifier effect on voltage comparator performance has been examined via the hierarchical synthesis approach, whose general framework is given in **Figure 6.13**. The synthesis flow is the same as the proposed hierarchical approach for cascaded amplifier. The resultant PoFs of both sub-blocks are combined by cross-product, and non-dominated solutions which are selected from the entire solution pool, create system-level PoF. Circuit simulation is utilized to measure some performance values such as drain-to-bulk voltage, and trans-conductance of input transistor, which are not considered in the optimization loop, they are just necessary to perform analytical analysis.

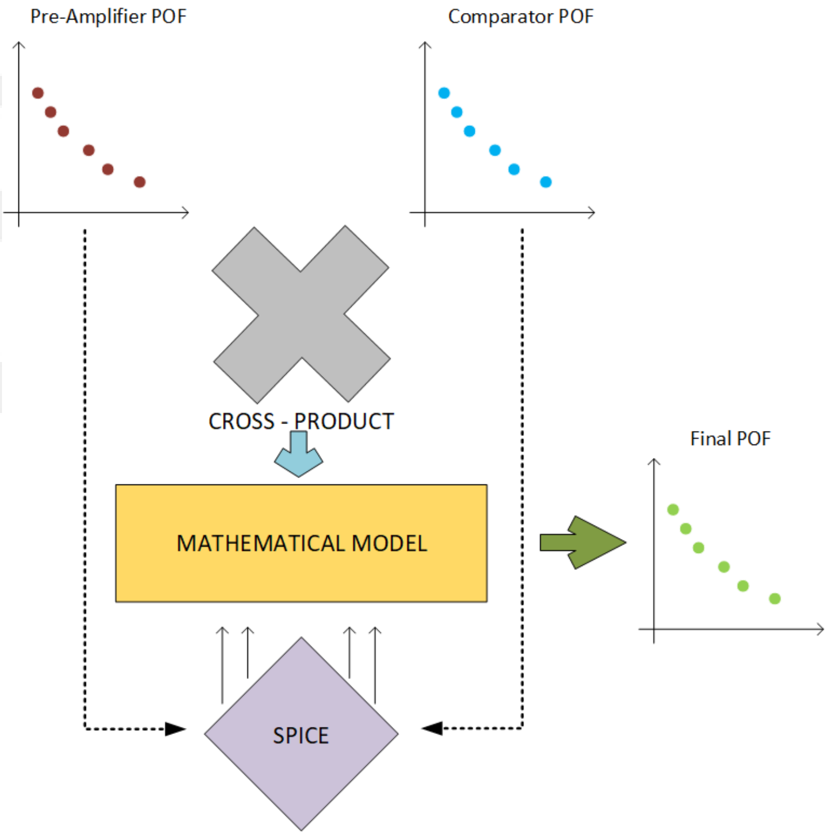


Figure 6.13: General framework of the hierarchical synthesis approach for comparator.

Before explaining the analytical modeling approach, it would be useful to make the necessary explanations for notation. Considering the synthesized comparator circuit, it can be seen that the latch comparator circuit is driven by an inverter that drives a load. As seen in the performance graph example in **Figure 6.14**, the latch output changes its state as a result of a small change in the input voltage (ΔV_i). On the other hand, the change in the latch output causes a state change in the comparator output. Here, the inverter's delay along with the latch's delay must be added to the comparator's delay. In the notation used in the following equations, $t_{D,inv}$ refers to the inverter delay, $t_{D,latch}$ refers to the latch delay, and $t_{D,comp}$ refers to the total delay of the comparator.

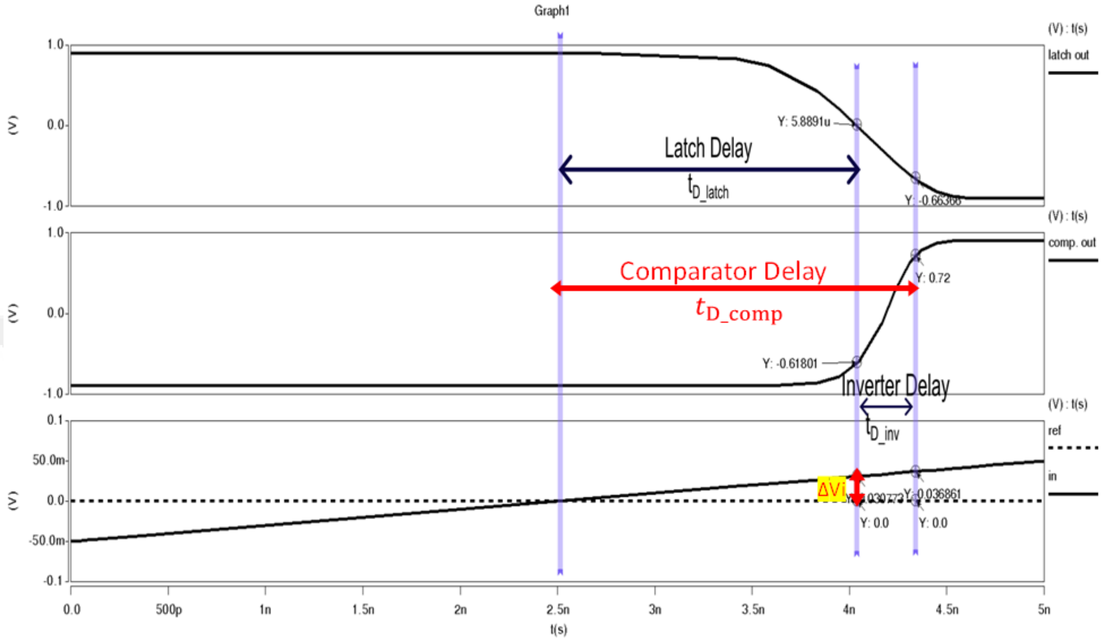


Figure 6.14: Delays of the synthesized voltage comparator.

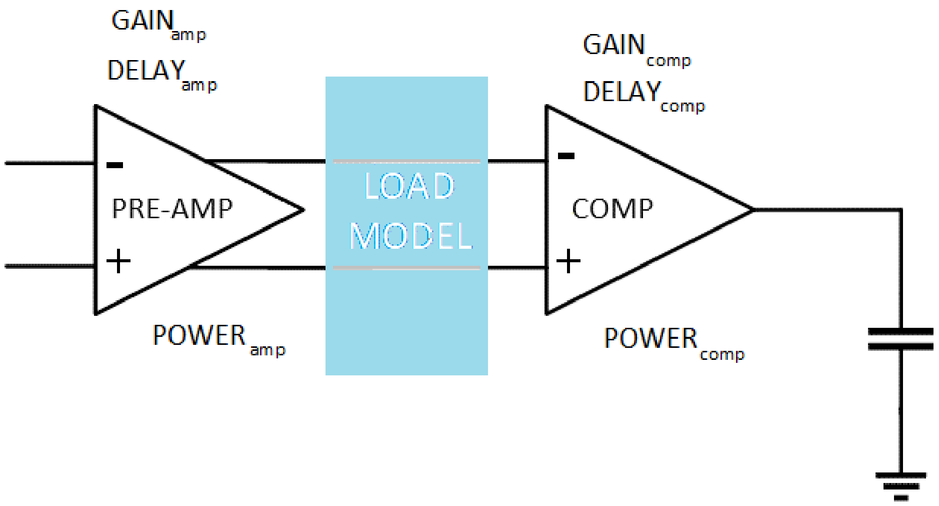


Figure 6.15: Representation of load model for the comparator with pre - amplifier.

An illustration of the combination of pre-amplifier and voltage comparator is shown in **Figure 6.15**, where the load model consists of a bunch of analytical equations. The time delay of a single-pole circuit can be expressed as in Equation 6.10, where V_{DD} is the supply voltage, ΔV_i is the minimum input voltage difference required for state change at the latch output, and τ_{comp} is the time constant of the circuit. Since all values except the time constant are known in this equation, the equation can be rearranged as in Equation 6.11 to find the time constant.

$$t_{D,latch} = \tau_{comp}, ln\left(\frac{2V_{DD}}{\Delta V_i}\right) \tag{6.10}$$

$$\tau_{comp} = \frac{t_{D,latch}}{ln(\frac{2V_{DD}}{\Delta V_i})} \tag{6.11}$$

As the input resolution of the comparator is increased by the gain of the pre-amplifier, latch delay is expected to decrease which also can be interpreted with the help of Equation 6.12. Also, the inverter delay is added to the new delay of the comparator block by Equation 6.13, to estimate the total delay of the comparator in the presence of the pre-amplifier.

$$t_{D,latch,new} = \tau_{comp}.ln\left(\frac{2V_{DD}}{\Delta V_i.A_{v,amp}}\right) \tag{6.12}$$

$$t_{D,comp,new} = t_{D,latch,new} + t_{D,inv,comp} \tag{6.13}$$

As a quick prediction, the way to achieve the best speed performance is to constantly increase the voltage gain of the pre-amplifier. However, it is not healthy to make such an inference only by considering Equation 6.12. Because, as seen in Equation 6.14, the time delay of the pre-amplifier is added to the time delay of the total system. Additionally, the total power consumption is calculated by summing the power consumption of two separate blocks like in Equation 6.15.

$$t_{D,tot} = t_{D,amp} + t_{D,comp,new} \tag{6.14}$$

$$power_{tot} = power_{amp} + power_{comp} \tag{6.15}$$

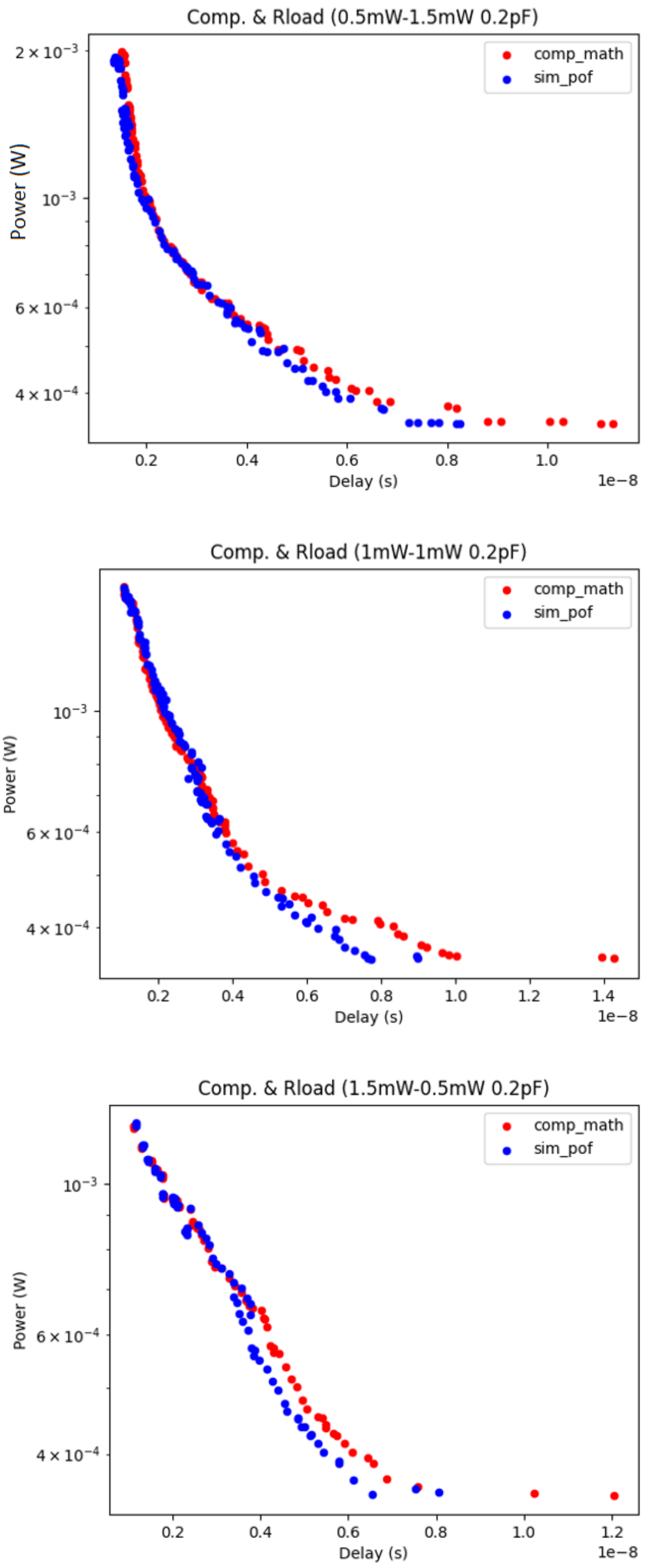


Figure 6.16: Obtained PoFs through the hierarchical synthesis of the comparator.

To conclude mathematical modeling, one step remains which is determining of amplifier delay. To perform delay calculation for amplifier a time constant is needed. The time constant of the pre-amplifier is expressed with Equation 6.16, where K is a constant used to improve the accuracy of amplifier delay estimation, $R_{out,amp}$ is the output resistance of amplifier calculated by Equation 6.17, $C_{tot,amp}$ is the self-capacitance at the output node of amplifier estimated via Equation 6.18, and $C_{tot,comp}$ refers to the total capacitance at the input of comparator that can be calculated by Equation 6.19.

$$t_{D,amp} = K.R_{out,amp}.(C_{tot,amp} + C_{tot,comp}) \tag{6.16}$$

$$R_{out,amp} = r_{o1}//R_L \tag{6.17}$$

$$C_{tot,amp} = C_{db,p} \tag{6.18}$$

$$C_{tot,comp} = C_{gs,n} + C_{gd,n} \tag{6.19}$$

The PoFs of the hierarchical design are given in **Figure 6.16**. To obtain PoF for the system block, non-dominant solutions are selected from all solutions formed by cross-multiplication in the design space. The accuracy of the model was evaluated through the comparison of PoFs obtained via simulation and model. As seen from the figures, the solution set formed by the mathematical model successfully captures the true PoF.

The effect of the voltage gain of the pre-amplifier circuit on the comparator delay can be seen in **Figure 6.17**. It is understood that the pre-amplifier gain does not need to be as large as possible to increase the speed performance of the voltage comparator. In fact, it is noteworthy that the amplifier uses low-gain individuals in the Pareto optimal set to achieve low delay at the system level. It is thought that two things may cause this situation. Firstly, the high output resistance of high-gain amplifiers increases the amplifier delay. Secondly, some difficulties in the comparator test bench may prevent precise measurements and predictions.

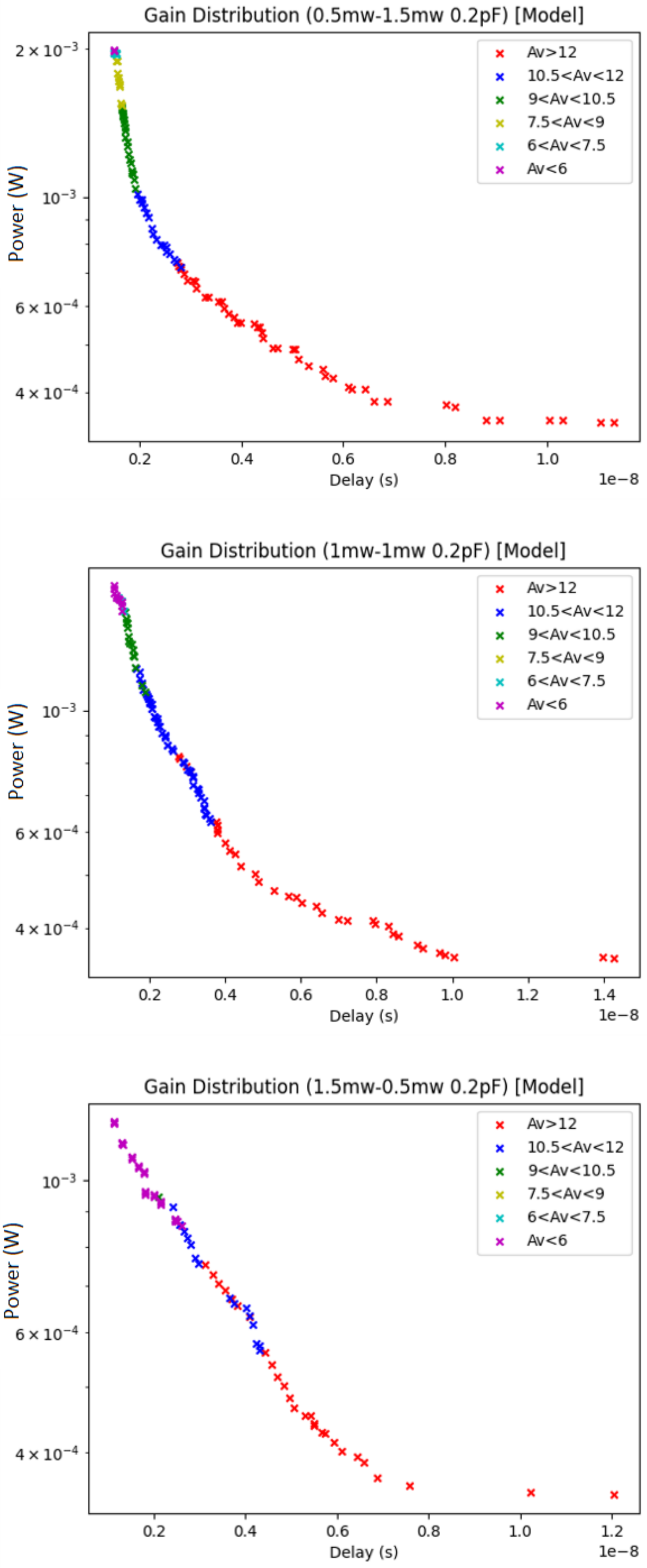


Figure 6.17: Gain distribution of preamplifier in obtained PoFs.

7. CONCLUSION AND FUTURE WORK

The idea behind the automation of analog/RF integrated circuits dates back several decades. Many Electronic Design Automation (EDA) tools have been developed over the years to automate all stages from design to deployment of integrated circuits. The first developed tools that automatically design circuits of limited complexity. After the first automation attempts, the most important development can be said the use of highly effective automation algorithms in analog/RF optimization problems. Thanks to effective optimization algorithms, development has moved towards circuit simulation-based optimization to increase accuracy. Although the success of simulation-based optimization methods has been proven, the search for methods to solve more complex circuits has continued. As a result of the creation of more stable circuit models and well-established automation methods, the optimization of quite complex analog/RF building blocks has been successfully achieved. Recently, thanks to powerful computing, machine learning methods are frequently used in the synthesis of analog/RF circuits. Despite all of these developments, the Analog/Rf IC automation problem has not yet been fully solved. Developers continue to try to balance the trade-off between effectiveness and efficiency.

Although analog/RF design automation takes part in many phases from design to production, it can generally be examined under two major headings; circuit sizing and layout generation. In circuit sizing optimization, the circuit is designed at the schematic level through an automation tool. On the other side, automation of layout generation deals with different problems with layout considerations such as floor planning, routing, and placement. This thesis focuses on circuit sizing automation via the developed multi-objective optimization tool to synthesize important analog/RF building blocks. To accelerate the automation process, artificial neural network models for estimating the behavior of building blocks have been developed. Additionally, some approaches are presented to enable the developed tool to gain the ability to perform design automation at the hierarchical level.

The analog/RF sizing problem has been fully automated thanks to the developed tools presented in the literature. However, most of the studies presented evaluate the effectiveness of the tool they developed with only a few case study examples. For this reason, a clear conclusion cannot be reached regarding the capacity of the developed tools to solve any complex and tight-constrained sizing problem. To palliate this problem, a tool has been developed which is capable to design of different analog/RF building blocks, automatically. The main analog/RF sub-blocks have been synthesized through this developed tool, where NSGA-II, a genetic algorithm, searches for optimal solutions. Obtained synthesis results have shown that circuit sizing has been accomplished via this tool, even if the circuit is highly nonlinear.

After the effectiveness of the automation tool has been proved, the developed simulation-based tool was modified to perform model-based optimization with the help of ANN models. The presented modeling and training data generation methods have increased the effectiveness of circuit models and helped the synthesis process become more efficient. Especially, by modeling highly complex circuits, complex systems/circuits can be optimized within practical time limits.

Circuits synthesized through simulation-based optimization were combined at a higher level with the help of mathematical equations to obtain system-level designs.The effectiveness of the proposed hierarchical optimization approach has been proven through simulations. Thus, it is thought that the way has been paved for the hierarchical optimization of complex analog/RF integrated circuits.

In summary, this thesis has three main titles; an optimization tool to solve multi/many-objective analog/RF optimization problem, modification of tool to accelerate optimization process, and hierarchical synthesis approach to synthesize analog circuits at the system level. As seen from this outline, it is aimed to effectively perform the hierarchical synthesis of complex analog/RF circuits. Therefore, as a first future work, a hierarchical synthesis tool should be strengthened to perform fully automated hierarchical synthesis. Circuit models may be created by the ML-based tool to represent each sub-block to realize bottom-up hierarchical synthesis. To reach a higher block hierarchically, sub-block models may be utilized to create the upper block with the help of a hardware description language such as Verilog-A, Verilog-AMS. At the next level, the performance of the obtained solutions for the upper-level block may be evaluated with the help of SPICE and the upper-level system can be modeled with all the data obtained. By continuing this loop until it reaches the highest level, the system-level synthesis may be carried out efficiently with little simulation effort.

As a second future work, a top-down hierarchical synthesis tool needs to be developed. Synthesis may start from the top level where system requirements are defined and continue down to lower levels with the help of hardware description languages. In top-down hierarchical optimization, it is very important to effectively perform circuit modeling with hardware description languages. To achieve this, two layers connected to each other in the hierarchy might be modeled via ANN, so that the relationship between layers can be predicted accurately. It is worth noting that although this method is effortless in terms of simulation, the hardware description languages may not be enough to perform highly accurate modeling. So the need to use an ANN model between two layers may appear in the process, and the hierarchical synthesis process may require serious effort in the modeling phase.

Finally, as the third and last future work, the whole hierarchical synthesis may be carried out by a constructed ANN model. It should not be forgotten here that a large data set may be needed to model the behavior of the entire system. Since the performances of the system-level circuit will be measured through simulation to create the dataset, as an outcome a serious simulation effort might be required. However, as the system-level model was created once, the same system can be easily synthesized with different sub-blocks in different topologies using hardware description languages without the need for any simulation.

REFERENCES

[1] Harjani R., Rutenbar R. A., Carley L. R., (1989), "OASYS: A Framework for Analog Circuit Synthesis", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 8(12), 1247–1266.

[2] Makris C. A., Toumazou C., (1995), "Analog IC design automation. II. Automated circuit correction by qualitative reasoning", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 14(2), 239–254.

[3] Degrauwe M. G., Nys O., Dijkstra E., Rijmenants J., Bitz S., Goffart B. L., Vittoz E. A., Cserveny S., Meixenberger C., Van Der Stappen G., et al. (1987), "IDAC: An interactive design tool for analog CMOS circuits", IEEE Journal of Solid-State Circuits, 22(6), 1106–1116.

[4] Koh H. Y., Sequin C. H., Gray P. R., (1990), "OPASYN: A compiler for CMOS operational amplifiers", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 9(2), 113–125.

[5] Del Mar Hershenson M., Boyd S. P., Lee T. H., (1998), GPCAD: A tool for CMOS op-amp synthesis, in "Proceedings of the 1998 IEEE/ACM International Conference on Computer-Aided Design", 296–303.

[6] Gielen G. G., Walscharts H. C., Sansen W. M., (1990), "Analog circuit design optimization based on symbolic simulation and simulated annealing", IEEE Journal of Solid-State Circuits, 25(3), 707–713.

[7] Kirkpatrick S., Gelatt Jr C. D., Vecchi M. P., (1983), "Optimization by simulated annealing", Science, 220(4598), 671–680.

[8] Kruiskamp W., Leenaerts D., (1995), DARWIN: CMOS opamp synthesis by means of a genetic algorithm, in "Proceedings of the 32nd annual ACM/IEEE design automation conference", 433–438.

[9] Paulino N., Goes J., Steiger-Garção A., (2001), Design methodology for optimization of analog building blocks using genetic algorithms, in "ISCAS 2001. The 2001 IEEE International Symposium on Circuits and Systems", Vol. 5, IEEE, 435–438.

[10] Nye W., Riley D. C., Sangiovanni-Vincentelli A., Tits A. L., (1988), "DELIGHT. SPICE: An optimization-based system for the design of integrated circuits", IEEE

Transactions on Computer-Aided Design of Integrated Circuits and Systems, 7(4), 501–519.

[11] Torralba A., Chavez J., Franquelo L. G., (1996), "FASY: A fuzzy-logic based tool for analog synthesis", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 15(7), 705–715.

[12] Zadeh L. A., (1988), "Fuzzy logic", Computer, 21(4), 83–93.

[13] Phelps R., Krasnicki M., Rutenbar R. A., Carley L. R., Hellums J. R., (2000), "Anaconda: simulation-based synthesis of analog circuits via stochastic pattern search", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 19(6), 703–717.

[14] Alpaydin G., Balkir S., Dundar G., (2003), "An evolutionary approach to automatic synthesis of high-performance analog integrated circuits", IEEE Transactions on Evolutionary Computation, 7(3), 240–252.

[15] Back T., (1995), "Evolution Strategies 1: Variants and their computational implementation", Genetic Algorithms in Engineering and Computer, Science, 127–140.

[16] Krasnicki M., Phelps R., Rutenbar R. A., Carley L. R., (1999), MAELSTROM: Efficient simulation-based synthesis for custom analog cells, in "Proceedings of the 36th annual ACM/IEEE Design Automation Conference", 945–950.

[17] Silva J., Horta N., (2002), Genom: circuit-level optimizer based on a modified Genetic Algorithm kernel, in "2002 IEEE International Symposium on Circuits and Systems. Proceedings", Vol. 1, IEEE, I–I.

[18] McConaghy T., Palmers P., Gielen G., Steyaert M., (2007), Simultaneous multi-topology multi-objective sizing across thousands of analog circuit topologies, in "Proceedings of the 44th annual Design Automation Conference", 944–947.

[19] Hornby G. S., (2006), ALPS: the age-layered population structure for reducing the problem of premature convergence, in "Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation", 815–822.

[20] Deb K., Pratap A., Agarwal S., Meyarivan T., (2002), "A fast and elitist multiobjective genetic algorithm: NSGA-II", IEEE Transactions on Evolutionary Computation, 6(2), 182–197.

[21] Liu B., Fernández F. V., Gielen G., Castro-López R., Roca E., (2009), "A memetic approach to the automatic design of high-performance analog integrated circuits", ACM Transactions on Design Automation of Electronic Systems (TODAES), 14(3), 1–24.

[22] Pradhan A., Vemuri R., (2009), Efficient synthesis of a uniformly spread layout aware pareto surface for analog circuits, in “2009 22nd International Conference on VLSI Design”, IEEE, 131–136.

[23] Lourenço N., Horta N., (2012), GENOM-POF: multi-objective evolutionary synthesis of analog ICs with corners validation, in “Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation”, 1119–1126.

[24] Fakhfakh M., Cooren Y., Sallem A., Loulou M., Siarry P., (2010), “Analog Circuit Design Optimization Through the Particle Swarm Optimization Technique”, Analog Integrated Circuits and Signal Processing, 63, 71–82.

[25] Kennedy J., Eberhart R., (1995), Particle swarm optimization, in “Proceedings of ICNN’95-International Conference on Neural Networks”, Vol. 4, IEEE, 1942–1948.

[26] Vural R. A., Yildirim T., (2012), “Analog circuit sizing via swarm intelligence”, AEU-International Journal of Electronics and Communications, 66(9), 732–740.

[27] Gupta H., (2012), “Analog circuits design using ant colony optimization”, International Journal of Electronics, Computer and Communications Technologies, 2(3), 9–21.

[28] Dorigo M., Birattari M., Stutzle T., (2006), “Ant colony optimization”, IEEE Computational Intelligence Magazine, 1(4), 28–39.

[29] Liu B., Wang Y., Yu Z., Liu L., Li M., Wang Z., Lu J., Fernández F. V., (2009), “Analog circuit optimization system based on hybrid evolutionary algorithms”, Integration, 42(2), 137–148.

[30] Weber T., Noije W. A., (2012), “Multi-objective design of analog integrated circuits using simulated annealing with crossover operator and weight adjusting”, Journal of Integrated Circuits and Systems, 7(1), 7–15.

[31] Akbari M., Shokouhifar M., Hashemipour O., Jalali A., Hassanzadeh A., (2016), “Systematic design of analog integrated circuits using ant colony algorithm based on noise optimization”, Analog Integrated Circuits and Signal Processing, 86, 327–339.

[32] Sallem A., Benhala B., Kotti M., Fakhfakh M., Ahaitouf A., Loulou M., (2013), “Application of swarm intelligence techniques to the design of analog circuits: evaluation and comparison”, Analog Integrated Circuits and Signal Processing, 75, 499–516.

[33] Pereira P., Helena Fino M., Ventim-Neves M., (2014), “Optimal LC-VCO design

through evolutionary algorithms", Analog Integrated Circuits and Signal Processing, 78, 99–109.

[34] Kammara A. C., Palanichamy L., König A., (2016), "Multi-objective optimization and visualization for analog design automation", Complex & Intelligent Systems, 2, 251–267.

[35] Lberni A., Marktani M. A., Ahaitouf A., Ahaitouf A., (2021), "Efficient butterfly inspired optimization algorithm for analog circuits design", Microelectronics Journal, 113, 105078.

[36] "An automated topology synthesis framework for analog integrated circuits, author=Zhao, Zhenxin and Zhang, Lihong" (2020), IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 39(12), 4325–4337.

[37] Dash S., Joshi D., Sharma A., Trivedi G., (2018*a*), "A hierarchy in mutation of genetic algorithm and its application to multi-objective analog/RF circuit optimization", Analog Integrated Circuits and Signal Processing, 94, 27–47.

[38] Dash S., Joshi D., Trivedi G., (2018*b*), "Multiobjective analog/RF circuit sizing using an improved brain storm optimization algorithm", Memetic Computing, 10, 423–440.

[39] González-Echevarría R., Roca E., Castro-López R., Fernández F. V., Sieiro J., López-Villegas J. M., Vidal N., (2016), "An automated design methodology of RF circuits by using Pareto-optimal fronts of EM-simulated inductors", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 36(1), 15–26.

[40] Liao T., Zhang L., (2018), "Efficient parasitic-aware hybrid sizing methodology for analog and RF integrated circuits", Integration, 62, 301–313.

[41] Afacan E., (2019), "Inversion coefficient optimization based analog/RF circuit design automation", Microelectronics Journal, 83, 86–93.

[42] Afacan E., Dundar G., (2019), "A comprehensive analysis on differential cross-coupled CMOS LC oscillators via multi-objective optimization", Integration, 67, 162–169.

[43] Zitzler E., Laumanns M., Thiele L., (2001), "SPEA2: Improving the Strength Pareto Evolutionary Algorithm", Technical Report No: TIK Report 103, ETH Zurich, Computer Engineering and Networks Laboratory, Switzerland.

[44] Zhou R., Poechmueller P., Wang Y., (2022), "An Analog Circuit Design and Op-

timization System with Rule-Guided Genetic Algorithm”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 41(12), 5182–5192.

[45] İslamoğlu G., Çakıcı T. O., Güzelhan Ş. N., Afacan E., Dündar G., (2021), “Deep learning aided efficient yield analysis for multi-objective analog integrated circuit synthesis”, Integration, 81, 322–330.

[46] Lberni A., Marktani M. A., Ahaitouf A., Ahaitouf A., (2024), “Analog circuit sizing based on Evolutionary Algorithms and deep learning”, Expert Systems with Applications, 237, 121480.

[47] Sanabria-Borbón A. C., Soto-Aguilar S., Estrada-López J. J., Allaire D., Sánchez-Sinencio E., (2020), “Gaussian-process-based surrogate for optimization-Aided and process-variations-aware analog circuit design”, Electronics, 9(4), 685.

[48] Du S., Liu H., Yin H., Yu F., Li J., (2021), “A local surrogate-based parallel optimization for analog circuits”, AEU-International Journal of Electronics and Communications, 134, 153667.

[49] Liao T., Zhang L., (2020), “Efficient parasitic-aware gm/ID-based hybrid sizing methodology for analog and RF integrated circuits”, ACM Transactions on Design Automation of Electronic Systems (TODAES), 26(2), 1–31.

[50] Wasserman P. D., (1993), “Advanced methods in neural computing”, John Wiley & Sons, Inc.

[51] Budak A. F., Gandara M., Shi W., Pan D. Z., Sun N., Liu B., (2021), “An efficient analog circuit sizing method based on machine learning assisted global optimization”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 41(5), 1209–1221.

[52] Zhang S., Yang F., Yan C., Zhou D., Zeng X., (2021), “An efficient batch-constrained bayesian optimization approach for analog circuit synthesis via multi-objective acquisition ensemble”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 41(1), 1–14.

[53] Lyu W., Xue P., Yang F., Yan C., Hong Z., Zeng X., Zhou D., (2017), “An efficient bayesian optimization approach for automated optimization of analog circuits”, IEEE Transactions on Circuits and Systems I: Regular Papers, 65(6), 1954–1967.

[54] Touloupas K., Sotiriadis P. P., (2021), “LoCoMOBO: A Local Constrained Multiobjective Bayesian Optimization for Analog Circuit Sizing”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 41(9), 2780–2793.

[55] Holland J. H., (1992), “Genetic algorithms”, Scientific American, 267(1), 66–73.

[56] Blank J., Deb K., (2020), “Pymoo: Multi-objective optimization in python”, Ieee Access, 8, 89497–89509.

[57] Boussaïd I., Lepagnot J., Siarry P., (n.d.), “A survey on optimization metaheuristics”.

[58] Zhang Q., Li H., (2007), “MOEA/D: A multiobjective evolutionary algorithm based on decomposition”, IEEE Transactions on Evolutionary Computation, 11(6), 712–731.

[59] Lampinen J., Storn R., (2004), Differential evolution, in “New optimization techniques in engineering”, Springer, 123–166.

[60] Teodorovic D., Dell’Orco M., (2005), “Bee colony optimization–a cooperative learning approach to complex transportation problems”, Advanced OR and AI Methods in Transportation, 51, 60.

[61] Bean J. C., (1994), “Genetic algorithms and random keys for sequencing and optimization”, ORSA Journal on Computing, 6(2), 154–160.

[62] Passino K. M., (2002), “Biomimicry of bacterial foraging for distributed optimization and control”, IEEE Control Systems Magazine, 22(3), 52–67.

[63] Hofmeyr S. A., Forrest S., (2000), “Architecture for an artificial immune system”, Evolutionary Computation, 8(4), 443–473.

[64] Sağlıcan E., Afacan E., (2023), “MOEA/D vs. NSGA-II: A Comprehensive Comparison for Multi/Many Objective Analog/RF Circuit Optimization Through A Generic Benchmark”, ACM Transactions on Design Automation of Electronic Systems, 29(1), 1–23.

[65] Banba H., Shiga H., Umezawa A., Miyaba T., Tanzawa T., Atsumi S., Sakui K., (1999), “A CMOS bandgap reference circuit with sub-1-V operation”, IEEE Journal of Solid-State Circuits, 34(5), 670–674.

[66] Audet C., Bigeon J., Cartier D., Le Digabel S., Salomon L., (2021), “Performance indicators in multiobjective optimization”, European Journal of Operational Research, 292(2), 397–422.

[67] Riquelme N., Von Lücken C., Baran B., (2015), Performance metrics in multi-objective optimization, in “2015 Latin American Computing Conference (CLEI)”, IEEE, 1–11.

[68] Yıldırım B., Kaya S., Afacan E., Dündar G., (2022), ”An Efficient Hierarchical Approach for Synthesis of Multi-Stage Wide-Band Amplifiers”, in “18th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)”, IEEE, 1–4.

BIOGRAPHY

He received bachelor's degree from Kocaeli University in 2021. He is currently pursuing master education in Gebze Technical University and he works as research and teaching assistant in the same organization. His main research topics are analog/RF circuit optimization, circuit modeling via artificial neural network, and analog/RF circuit design.

PUBLICATIONS AND PRESENTATIONS FROM THE THESIS

[1] Sağlıcan, E., Afacan, E. (2023), ”MOEA/D vs. NSGA-II: A Comprehensive Comparison for Multi/Many Objective Analog/RF Circuit Optimization Through A Generic Benchmark” ACM Transactions on Design Automation of Electronic Systems, 29(1), 1-23.

[2] Sağlıcan, E., Dur, B., Afacan, E., (2023), ”Design of Low Power Low Noise On-Chip BioAmplifier in Cooperation with Analog IC Synthesis at 130nm Skywater Technology”, 19th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD), Funchal, Portugal, 3-5 July.

[3] Bayram, A., Sağlıcan, E., Afacan, E., (2023), ”A Comprehensive Performance Space Comparison of Bandgap Reference Circuits Through Hard Constrained Analog Circuit Multi-Objective Optimization”, 14th INTERNATIONAL CONFERENCE on ELECTRICAL and ELECTRONICS ENGINEERING (ELECO), Bursa, Türkiye (online event), 30 November - 2 December.

[4] Taşkıran H., Hacımustafaoğlu F. E., Sağlıcan E., Afacan E., (2023), ” ANN-Powered Reinforcement Learning-Based Analog Circuit Optimization”, 30th IEEE International Conference on Electronics, Circuits and Systems (ICECS), İstanbul, Türkiye, 4 - 7 December.

[5] Sağlıcan E, Bayram A., Afacan E., (2023), ” Two-Archive Evolutionary (TAEA)-Based Multi&Many Objective Analog IC Optimization”, 30th IEEE International Conference on Electronics, Circuits and Systems (ICECS), İstanbul, Türkiye, 4 - 7 December.