NATIONAL UNIVERSITY OF SCIENCE AND TECHNOLOGY POLITEHNICA BUCHAREST

Doctoral School of Electronics, Telecommunications and Information Technology

Decision No. 182 **from** 13-09-2024

Ph.D. THESIS SUMMARY

Cătălin-Laurențiu VIȘAN

TEHNICI DE INTELIGENȚĂ ARTIFICIALĂ PENTRU AUTOMATIZAREA PROCESULUI DE DIMENSIONARE A CIRCUITELOR INTEGRATE

ARTIFICIAL INTELLIGENCE TECHNIQUES FOR INTEGRATED CIRCUIT DESIGN AUTOMATION

THESIS COMMITTEE

Prof. Dr. Ing. Mihai CIUC National University of Science and Technology Politehnica Bucharest	President
Prof. Dr. Ing. Corneliu BURILEANU National University of Science and Technology Politehnica Bucharest	PhD Supervisor
Acad. Prof. Dr. Ing. Horia-Nicolai TEODORESCU „Gheorghe Asachi" Technical University of Iași	Referee
Prof. Dr. rer. nat. Georg PELZ Infineon Technologies AG & University of Duisburg-Essen	Referee
Conf. Dr. Ing. Horia CUCU National University of Science and Technology Politehnica Bucharest	Referee

BUCHAREST 2024

Table of contents

Chapter 1

Introduction

1.1 Presentation of the field of the doctoral thesis

Figure 1.1 outlines the initial steps of the analog circuit development flow. The process begins with a list of circuit specifications defined by the circuit architect or application engineer, based on the desired functionality. The designer then selects an appropriate circuit topology. The next step, circuit sizing, involves determining the optimal values for the circuit components to meet the functional constraints specified and enhance overall performance. A shallow verification is typically performed during this stage to ensure basic robustness of the design. The verification stage starts with a comprehensive performance evaluation, considering various environmental variables. If the circuit fails under certain conditions, the sizing process is revisited to address these failures.

Once the circuit passes verification, the layout is manually generated. At this point, the circuit's performance is re-evaluated, taking parasitic effects from the layout into account. If these parasitic effects significantly degrade performance, adjustments to the layout are necessary. Once the circuit meets the performance and robustness criteria, it moves into production. After fabrication, post-silicon verification is conducted to validate the circuit's behavior under real-world conditions, ensuring it meets the original design specifications.

With the advent of Artificial Intelligence, there has been increasing interest in automating analog circuit design processes. Consequently, recent research has extensively explored various design tasks such as pre-silicon verification [1, 2], layout generation [3], and post-silicon verification [4]. Moreover, circuit modelling [5, 6] has been employed to enhance the reusability of the designs. This thesis addresses the task of circuit sizing, and it incorporates the shallow verification process typically associated with this stage.

The norm in circuit sizing is that highly skilled engineers choose the suitable values for the circuit components. This activity requires manual effort and keeps the circuit designer busy for a significant amount of time. Even though manual circuit sizing is an iterative task, it cannot be easily automated since it is highly dependent on the designer's

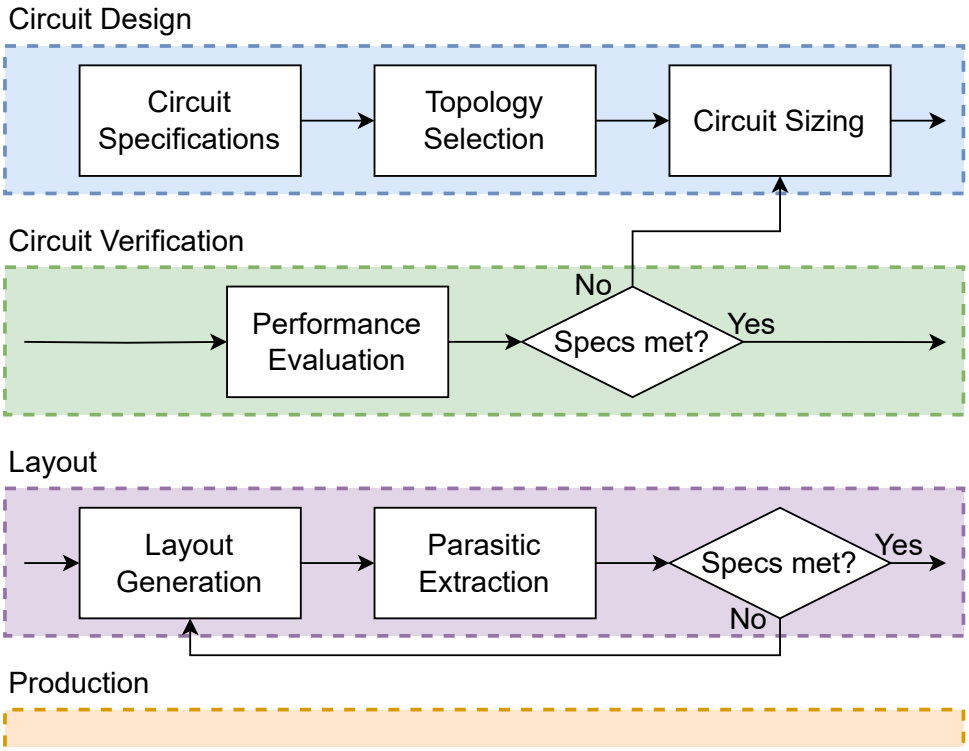

Fig. 1.1 Stages of the analog circuit design flow

experience. This task became a popular challenge in Electronic Design Automations and numerous techniques have been proposed over the years. The goal of these methods is to reduce the designers' effort, letting them focus on more creative tasks. As a result, the productivity and employee satisfaction in the semiconductors companies can be positively impacted.

1.2 Scope of the doctoral thesis

The automated circuit sizing techniques can be divided into qualitative and quantitative approaches. One qualitative method is rule-based design [7]. It involves creating algorithms and equations based on designers' prior knowledge. This method provides some degree of automation, but it still requires significant effort from the designer. Another class of qualitative methods is model-based sizing. They use simplified model equations, so they are quick and reusable. One example is Geometric Programming [8–10] where circuit responses are expressed as posynomial equations of the design parameters. Another example is Semidefinite Programming (SDP) relaxations [11]. It uses regression models based on simulation data to express the circuit responses. Even though they have important advantages, model-based sizing methods are unreliable due to the model's intrinsic deviation from the real circuit. [12]

Quantitative approaches rely on circuit simulations. As a result, they maintain fidelity to the real circuit without introducing deviations. The circuit is formulated as a constrained multi-objective optimization problem, with design parameters serving as inputs and circuit responses as outputs. The most commonly used methods to address this problem are evolutionary algorithms (EAs) [13]. They are stochastic global searching engines that mimic the survival of the fittest natural processes. Other quantitative approaches used in automated circuit sizing are Simulated Annealing (SA) [14, 15] and Particle Swarm Optimization (PSO) [16, 17]. The drawback of these methods is the high number of circuit simulations required.

To address the shortcomings of pure quantitative approaches, hybrid methods have been developed. They combine the global searching power of EAs with model-based circuit sizing to reduce the number of circuit simulations required. Due to the increased availability of computing power in the last two decades, Machine Learning (ML) models became a popular choice. Some early examples are focused on radio-frequency integrated circuits design [18, 19]. However, during the time these methods were employed in a broader range of applications, with recent developments being general purpose analog circuit sizing methods [20, 21].

The two aspects that have to be taken into consideration while discussing ML enhanced EAs are the model and the algorithm. A significant part of the proposed solutions [19, 21, 22] use Gaussian Processes [23, 24] as ML model. Alternatively, some recent works try to integrate neural networks [20] to reduce the computational costs of Gaussian Processes. However, it is not a straight forward path since neural networks need many more training points to work properly. In terms of the algorithms used, many works [19, 20, 22] are based on Differential Evolution [25]. Other methods are based on genetic algorithms [26–28], specifically, the nondominated sorting genetic algorithm II (NSGA-II) [29, 30].

Even though ML enhanced EAs were thoroughly debated in the literature, there are some limitations that have to be addressed. Mainly, most of the circuit sizing applications require optimizing conflicting objectives. However, many methods are either focused on solving single objective problems [19, 31], or they use scalarization to deal with multiple objectives [32, 12]. Curiously, there are just a few true multi-objective ML enhanced EAs used for analog circuit sizing. One early proposal is GPOF-SVM [33] combining an NSGA-II based optimizer (GENOM-POF [26]) with Support Vector Machines.

While ML enhanced EAs are the most common approach for circuit sizing, alternative methods have been developed. An example that is gaining a lot of traction in recent years is Bayesian Optimization [34]. Instead of using the ML model to minimize the number of simulations required by an EA, Bayesian Optimization uses the model as a support for an optimizer. The optimizer searches for the model's optima as described by an acquisition function. Thus, the next evaluation points are the model's optima, instead of evolutions of the previous solutions.

An early popular Bayesian Optimization method for automated circuit sizing is WEIBO [12]. It is specially designed to handle constrained problems. However, it is not a true multi-objective approach since it uses Tchebysheff scalarization to construct a figure of merit function. A state-of-the-art Bayesian Optimization approach is MACE [35]. It also constructs a model based on a figure of merit function. Nevertheless, instead of a single acquisition function, it employs three of them, and their optimization is truly multi-objective. Its advantage is finding the trade-offs between acquisition functions, since none of them can outperform the others in all the problems [36].

Moving beyond the optimization algorithms, one commonly overlooked aspect in automated circuit sizing literature is the handling of operating corners. These refer to variations in process, voltage, and temperature (PVT) and are often called PVT corners. Typically, designers select the most unfavorable PVT corner in advance, leveraging their expertise. Going further, they perform circuit sizing for this condition and, in the end, the design is verified in all PVT corners to demonstrate its robustness. This minimizes the number of circuit simulations required for circuit sizing, but often leads to multiple design-verification loops because the solution which was feasible for the PVT corner chosen up-front turns out not to feasible in all other corners. Alternatively, the sizing can be done while verifying all PVT corners for each sizing configuration proposed by the algorithm. However, this is computationally expensive (ex. requires 27x more simulations for a usual case of 3 operating conditions with 3 levels each) and leads to exhausting the simulation budget without being able to find feasible solutions.

1.3 Content of the doctoral thesis

Given the current state of research in automated circuit sizing, there remains significant room for improvement in several areas. The primary goal of this thesis is to reduce the time required for circuit sizing compared to traditional methods. To accomplish this, we focus on two key aspects of the algorithm: efficiency and robustness. Within this context, we define the following specific objectives:

(a) Find the most promising evolutionary algorithm for circuit sizing tasks. While evolutionary computation underpins many automated circuit sizing methods, the algorithm selection in most studies appears arbitrary. Thus, a detailed analysis is required to identify the most suitable algorithm for this application.

(b) Develop a circuit sizing method which reduces the simulation budget required to find circuit configurations that meet the specifications by at least 50% compared to classic evolutionary algorithms.

(c) Explore Bayesian Optimization and Machine Learning enhanced evolutionary algorithms to build a robust algorithm with consistent performance across various circuits.

(d) Find alternative methods to reduce the simulation budget. Given the law of diminishing returns, continually developing more advanced optimization algorithms may not be the most efficient approach. Instead, reframing the problem definition could offer significant speedups in circuit sizing.

(e) Include more verification within the automated circuit sizing process to minimize the need for repeated design-verification cycles.

To address these objectives, the rest of the thesis is structured as follows:

We begin by providing the technical background, literature review, and circuit descriptions in Chapter 2. The technical background (Section 2.1) includes the formulation of the problem, the description of the metrics used, and detailed descriptions regarding key concepts like evolutionary algorithms, Gaussian Processes and Bayesian Optimization. The literature review (Section 2.2) presents the current state-of-the-art in automated circuit sizing.

We compare evolutionary algorithms (EAs) in Chapter 3 to identify the most promising one for circuit sizing tasks, as stated in objective (a). We start from five state-of-the art EAs, and we use synthetic benchmarks to find the best hyperparameter configurations. Then we apply the algorithms on two optimization tasks targeting real circuits, specifically voltage regulators. While the first circuit has more design variables (27, as opposed to 8), the second one has more objectives to be optimized (6, instead of 3). Apart from the performance assessment, we also analyze diversity of the solutions offered by the various algorithms and draw conclusions regarding which of them brings more benefits to the circuit designer.

In Chapter 4 we start to address the shortcomings of pure quantitative approaches using hybrid methods. Specifically, we introduce a novel Machine Learning (ML) enhanced EA to speed up the optimization process, in line with objective (b). It is a true multi-objective optimization algorithm which combines the most promising EA discovered in Chapter 3 with Gaussian Processes. It is designed to work well on problems with many variables, objectives, and constraint specifications. As part of its development, we propose innovative population survival policies and offspring selection algorithms that reduce the number of real simulations required to complete the optimization process and preserve the solution diversity. To the best of our knowledge, the proposed method is the first to specifically address solution diversity by searching directly over the multi-objective space using Pareto dominance.

Treating the problem of circuit sizing in a true multi-objective manner is one of the main pillars of the methods proposed in this thesis. Also, as opposed to previous work, the circuits on which we evaluate our methods have numerous design variables and responses that should satisfy constraints. Most research in this area has performed tests on circuits with fewer than 10 design variables, and even though very recent methods

[37] considered problems with 24 and 20 design variables, they only had one target objective and six or two constraints.

In Chapter 5 we go further into hybrid methods focusing on Bayesian optimization. More precisely, we present an original framework for automatic circuit sizing which combines ML-enhanced EAs with Bayesian optimization, in accordance with objective (c). This novel strategy is called Evolutionary Bayesian Optimization (EBO). On one hand it follows the ML enhanced EAs strategy of global search using evolutionary computation, which was proven to be effective. On the other hand, as in Bayesian Optimization, our technique fully takes advantage of the powerful ML model, by searching for its optima instead of using it just as a preselector. Moreover, our method performs a truly multi-objective optimization, and it uses population-based search that takes advantage of parallel simulation infrastructure.

The concept of PVT corners management is presented in Chapter 6. We address the shallow PVT verification — specific to the circuit sizing phase — in an intelligent manner, by employing operating corners management strategies during the optimization (objective (d)). We propose three different strategies, and we also combine two of them in order to obtain maximum efficiency. First strategy is a two-steps approach (Section 6.1), where the algorithm identifies promising sizing candidates first by considering solely the nominal corner. Then, as the second step, it begins to take into account extreme operating conditions. This simple technique leads to fewer circuit simulations spent on unpromising candidates during the early stages of optimization. The second strategy is a periodic worst corners selection mechanism (Section 6.2). It allows obtaining similar results as if optimizing in all corners while drastically reducing the simulation budget. The two strategies can be combined in order to improve efficiency even further (Section 6.3).

The third corners management strategy employs a prioritized list of corners, ordered by their difficulty level (Section 6.4). Depending on the use-case, the list might include the nominal corner or not. Solutions undergo sequential evaluation across these corners, with the assessment of each solution halting as soon as it fails to meet the specifications for a given corner. Consequently, this approach minimizes the simulations expended on lower-quality solutions, concentrating the optimization efforts on the more challenging corners. Moreover, this approach enables the circuit sizing algorithm to consider many PVT corners, which can lead to a reduction in the number of design-verification loops needed to obtain a robust circuit, as outlined in objective (e). This is a critical point in improving the adoption rate of automated circuit sizing methods since it assures a good time to market for new semiconductors products.

Finally, in Chapter 7 we draw the conclusions and discuss the limitations of our research. In addition, we go through some of the avenues that can be explored in the future.

Chapter 2

Background

2.1 Technical Brief

2.1.1 Problem formulation

In circuit design, highly skilled engineers select a specific topology and then manually tune design parameters to meet certain specifications and, ideally, enhance overall performance. Typically, design parameters encompass resistor and capacitor values, along with transistor parameters like multiplicity, channel length, and width. The specifications refer to performance measures, known as circuit responses, that must meet certain thresholds. For instance, the specifications of low-dropout voltage regulators might include phase margin, power supply rejection ratio (PSRR), and settling time. Additionally, certain circuit responses require optimization beyond meeting specifications, such as minimizing chip area and reducing current consumption.

Therefore, circuit sizing can be formulated as a constrained multi-objective optimization problem (Equation 2.1).

$$\begin{aligned} &\text{minimize } f_i(X),\ i \in N_{objectives} \\ &\text{subject to} \\ &g_j(X) < \text{treshold}_j,\ j \in N_{specifications} \end{aligned} \tag{2.1}$$

where X is the input vector of design parameters, $g_j(X)$ are the constraints representing the circuit specifications, and $f_i(X)$ are the objectives representing the circuit responses which require further optimization. It is worth noting that Equation 2.1 presents the case of a pure minimization problem. In practice, objectives and constraints that require maximization can be converted into minimization problems by negating their values (Equation 2.2).

$$\begin{aligned} &\text{maximize } f_i(x) \equiv \text{minimize } -f_i(x) \\ &g_j(X) > \text{treshold}_j \equiv -g_j(X) < -\text{treshold}_j \end{aligned} \tag{2.2}$$

Usually, the design parameters are integer or real numbers. However, sometimes enumerated types, stepped types, or even categorical variables are included. Also, it is worth noting that an objective $f_i(X)$ can also have an associated constraint $g_j(X)$. For example, the current consumption must be below a certain threshold, but it would be best to be as low as possible.

Constraint handling is a crucial aspect of circuit sizing. It involves satisfying all circuit specifications for a solution to be feasible. Constraint handling techniques are categorized into three main types [38]: those that prioritize feasible solutions to survive until the next generation, those that aim to balance the trade-off between feasibility and convergence, and those used for repairing infeasible solutions. This thesis focuses on the techniques in the first category because they are robust, well understood, and tested.

As a result, solving a constrained multi-objective optimization problem requires a two-step approach. First, the algorithm must find solutions that meet the specifications, termed as feasible solutions. Second, the algorithm must optimize the objectives. Thus, we consider an optimization run has two phases, feasible solutions search and objectives' optimization. This two-step strategy for constrained optimization is well established in the literature [39].

As with any multi-objective optimization problem, the optimum is not represented by a single point within the objectives' high-dimensional space (often referred to as hyperspace). Instead, it is represented by a Pareto front, which makes the optimization process even more difficult. The quality of an algorithm is measured by how well it depicts the Pareto front of the problem at the end of the optimization. On one hand, it is important to generate solutions as close as possible to the Pareto front. On the other hand, the solutions should be spread across the entire front, not just in a single region. Having solutions across the entire Pareto front is important in circuit sizing, because it gives the designer access to various trade-offs.

In practice, circuit sizing tasks deviate from classic multi-objective optimization problems, since the circuits are subjects to the operating conditions. Operating conditions are circuit inputs that remain outside the algorithm's control. They represent environment variables such as process corner, temperature, reference voltage etc. Similar to design parameters, operating conditions can be of various data types. A particular combination of operating conditions values represents an operating corner, often referred as process voltage temperature (PVT) corner. Usually, the relevant operating corners are set by the designers after performing Monte Carlo simulations. A design is considered robust if it fulfills the specifications across all operating corners, or lies within a certain tolerance range [40].

2.2 Literature Brief

Evolutionary Algorithms Evolutionary Algorithms (EAs) are stochastic global search methods that mimic the survival of the fittest natural processes. The generic flow diagram of EAs for circuit sizing tasks is presented in Figure 2.1. EAs differentiate themselves based on their survival policies and the strategies they employ for offspring generation. Most of the relevant developments in the field of automated circuit sizing are using genetic algorithms or Differential Evolution as their offspring generation engine. Survival policies across different EAs are notably varied. However, they can broadly be divided into two categories: those that adhere to elitism and those that promote diversity.

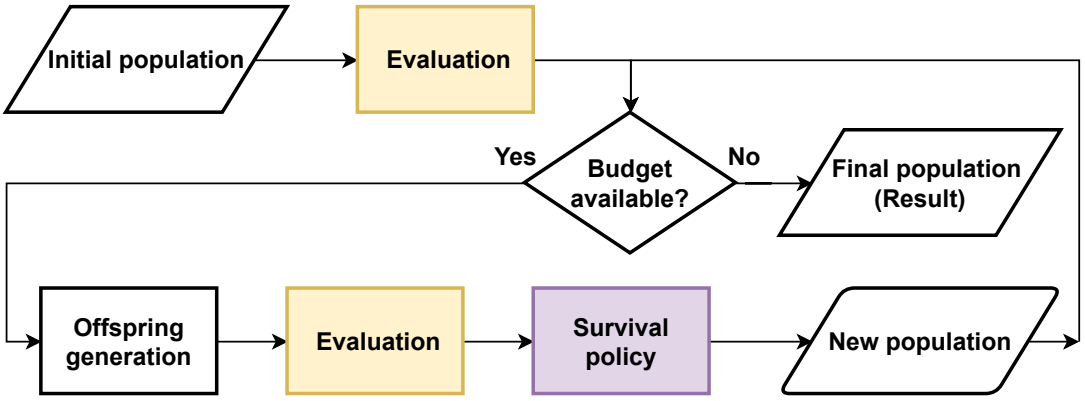

Fig. 2.1 Flow diagram of evolutionary algorithms for circuit sizing

There are many examples of successfully employing EAs in circuit sizing tasks. One of the most widely used algorithms is NSGA-II [30], which was used as-it-is for automatic analog integrated circuit sizing in [26, 27] or by adding a clustering method to reduce the number of simulations required [28]. Other popular algorithms employed for circuit sizing are Particle Swarm Optimization and various variants of differential evolution [41].

Surrogate Models Although EAs can produce good-quality designs, optimizing efficiency with respect to runtime is crucial. Simulations can be extremely computationally expensive; therefore, using default EAs can require an impractically long time to produce the required results. Hence, surrogate models are used to replace a fraction of expensive simulations. The construction and use of a surrogate model to predict the function values typically requires considerably less effort than embedding the expensive function evaluator within the optimizer as-is.

The most popular surrogate models used in the past several years are GPs and artificial NNs; however, some other ones, such as response surface methods and support vector machines [42] have also been used.

Chapter 3

Evolutionary Algorithms for Circuit Sizing

This chapter is a study on classic Evolutionary Algorithms used in the context of automated circuit sizing. It is mainly based on [43] and [44].

In the simulation-based methods, the circuit is treated as a black box and its optimization is based only on simulations. There are multiple metaheuristic algorithms suitable for circuit sizing tasks, but by far the most popular are the Evolutionary Algorithms. In this chapter we discuss the most promising state-of-the-art EAs in the context of circuit sizing. First, we use synthetic benchmarks to find the best hyperparameter configurations, and we apply the algorithms on a real low-dropout (LDO) voltage regulator. Our goal is to find the most promising EAs and to assess the impact of randomness in the circuit sizing task. Second, we extend the analysis to another LDO, changing the focus to population diversity and the versatility of the EAs.

3.1 Calibration on Synthetic Benchmarks

In an ideal scenario, the five EAs would be calibrated on a real and generic circuit sizing problem. Unfortunately, a single run representing a hyperparameter configuration of an algorithm can take more than a full day. Thus, the hyperparameters configurations used are the result of algorithms' calibration on synthetic benchmarks.

The experimental results in this subsection are obtained on the 9 synthetic benchmarks from the Walking Fish Group (WFG) toolkit [45]. This toolkit is scalable and supports any number of objectives. Moreover, it incorporates a variety of important characteristics that widely exist in real-world problems. To construct a test problem, it only requires to specify a shape function, which determines the Pareto front, and a transformation function, which describes the fitness landscape.

All optimizations on synthetic functions (each algorithm on each problem) were done 10 times with different random starting populations and random seeds, and results

averaged over these 10 runs. We have chosen 5 and 10 as the number of objectives, as we can compare how the different optimization algorithms perform on problems with fewer objectives (5) and with more objectives (10).

3.2 Evaluation on Real Circuits

In this subchapter, we evaluate the target algorithms on two voltage regulators. The focus of the comparison is not only performance, but also robustness and population diversity.

3.3 The Most Promising Algorithm for Circuit Sizing

The focus of this chapter is to evaluate the state-of-the-art EAs in terms of performance, versatility, and diversity of the population. In this context we understand versatility as the capacity of the algorithm to have similar performance on different circuits. The performance of GDE3 [46] is remarkable and steady across all the considered situations. Thus, we can argue that it is the most versatile algorithm amongst the five. It is true that the performance of IBEA [47] is also relatively good on the problems that we considered, but the impossibility of using it on problems with a high number of objectives makes the algorithm less versatile.

In terms of solution diversity, GDE3 has the best results, given its intrinsic capabilities and the potential of performing well with a large population. Overall, we conclude that GDE3 is the most promising EA for circuit sizing automation. Its performance is very good, it proved to be versatile, and it preserves the most diverse population amongst the state-of-the-art EAs. Sometimes its nature leads to a slower convergence, but in the circuit sizing applications, where the level of complexity of the problem can fluctuate heavily from a circuit to another, we consider a slower but more robust convergence desirable.

Evolutionary algorithms are good for sampling the hyperspace of the design parameters, but using them alone will result in applications that require a high number of simulations. The state-of-the-art solutions for circuit sizing use machine learning based surrogate models to reduce the total number of simulations. Therefore, we aim to build an efficient circuit sizing algorithm by combining GDE3 with such a surrogate model.

Chapter 4

A Machine-Learning enhanced Evolutionary Algorithm

This chapter is based on the following articles: [48] and [49]

Evolutionary algorithms (EAs) are powerful stochastic searching methods. However, in their original form, they require too many real circuit evaluations to discover optimal solutions. In this chapter, we pair arguably the most promising EA for circuit sizing tasks, GDE3 [46], with Gaussian Processes (GPs) to reduce the number of circuit simulations. We call the resulting sizing algorithm "Multi-objective Optimization based on Differential Evolution and Bayesian Inference" (MODEBI).

The main contributions of this chapter are as follows. First, we propose a multi-objective optimization method based on the combination of a GDE3 inspired algorithm with GPs that is designed to work well on problems with many variables, objectives, and constraint specifications. Second, we propose innovative population survival policies and offspring selection algorithms that reduce the number of real simulations required to complete the optimization process and preserve the solution diversity. Finally, the proposed method is evaluated, and its performance is compared with state-of-the-art EAs and Bayesian Optimization algorithms on two real voltage regulators.

4.1 Surrogate Modelling

Optimizing runtime efficiency is essential. To achieve this, surrogate models are employed to replace the real circuit simulator for a portion of the costly evaluations. Building and utilizing a surrogate model to predict function values generally requires significantly less computational effort than directly integrating the expensive function evaluator into the optimizer. Gaussian Processes and artificial neural networks have been the most widely used surrogate models in the literature. Recent approaches, including ours, utilize online training, where the model is incrementally updated during optimization with each new simulation. While the use of surrogate models adds some

computational overhead, they more effectively guide the selection of future candidate points, leading to faster convergence.

4.1.1 Using Gaussian Processes

The GP regression is based on Bayesian inference, wherein a prior statistical model can be combined with observed evidence to derive a more accurate statistical model. A prior distribution can be specified using prior mean and kernel functions. In the absence of prior knowledge, the mean is assumed to be zero. The kernel function represents the similarity measure between the function values at two locations. Training the GP over the available data points involves encoding the information in the mean and the kernel. For each new point, the GP will predict the mean (μ), representing the estimated function value, and the variance (σ^2), representing the prediction uncertainty. These predictions are refined incrementally when new data are observed.

An acquisition function is typically used to explore the state space using the GP model. Various types of acquisition functions can be used to balance the exploration and exploitation during the optimization. In the proposed algorithm, we employ the Lower Confidence Bound (LCB) as follows (Equation 4.1):

$$LCB(x) = \mu(x) - K\sigma(x) \tag{4.1}$$

In minimization problems, a larger K corresponds to a highly exploratory behavior.

4.2 Multi-Objective Optimization based on Differential Evolution and Bayesian Inference (MODEBI)

In this chapter we describe the MODEBI algorithm starting from its GDE3 inspiration. The chapter includes the different candidate selection (and survival) mechanisms and the resulting versions of MODEBI.

4.2.1 Machine Learning Enhanced Generalized Differential Evolution 3

Population Diversity Population diversity is a critical aspect of multi-objective evolutionary algorithms owing to the risk of ignoring certain areas of the input space and the goal of finding multiple points in the Pareto Front. The original GDE3 algorithm has a good mechanism of preserving the population diversity: it compares each offspring only with its parent to determine the appropriate solution for inclusion in the next generation. However, integrating the GP surrogate model introduces some changes in the original GDE3 approach, and additional incentive for diversity preservation is required.

The original GDE3 algorithm uses Crowding Distance [30] in the pruning phase. We instead applied the Distribution Metric [50] not only to monitor the diversity across the optimization, but also to select suitable individuals.

Preselection We employed the Gaussian Process models as a preselection mechanism. To maximize the potential benefit, the GDE3 offspring generator was used to generate ten times more offspring ($10N$) than the population size (N). Simultaneously, we aimed to reduce the number of simulations performed at each epoch to a quarter of the population size ($N/4$ instead of N). Thus, we adopted the surrogate model to select the most promising $N/4$ offspring from the $10N$ generated. By replacing at most a quarter of the population at each step, MODEBI maintains a good balance between exploration and exploitation. The exploratory behavior of the evolution is promoted by maintaining a high number of individuals in the population. Simultaneously, only the offspring with the best improvement rates are considered to join the next population.

One Gaussian Process is trained for each circuit response by using the design parameters and operating conditions as inputs. This approach incurs a lower computational cost and allows parallel training and querying of Gaussian Processes. However, it has a limitation in that it does not consider the correlations between responses, which can be addressed in future work. We use a scaled *radial basis function* kernel as it is a popular choice for ML, and its reduced complexity makes model training feasible on all available data points (up to $20,000$ in our test cases). Moreover, good accuracy can be achieved with a reasonable computational cost.

The most promising $N/4$ offspring (as predicted by the Gaussian Processes and selected through one of the selection functions introduced below) are evaluated using the circuit simulator. Finally, the new population is created by selecting the best individuals from the previous population and these offspring through one of the survival functions described below. The updated method is presented in Algorithm 1.

Algorithm 1: MODEBI for circuit sizing

Input: A random population **POP** of size **N**

```
while simulation budget available do
    GP = train(available simulations)
    offspring = 10×generate(POP, GDE3Operator)
    evaluate(offspring, GP)
    bestOffs = select(offspring, N/4)
    simulate(bestOffs)
    POP = survival(POP, bestOffs, N)
end while
```

Although generating $10N$ offspring and replacing only $N/4$ individuals in the population is beneficial, as explained above, it results in additional complexity. The original GDE3 offspring selection and survival procedures cannot be used without alterations.

Summary MODEBI can be regarded as an extension of GDE3 that incorporates Gaussian Process surrogate models for preselecting offspring to accelerate convergence without wasting time on real circuit simulations. The preselection must be performed carefully to preserve the population diversity and promote the most promising offspring.

4.3 MODEBI Evaluation

In this section we evaluate the MODEBI algorithm. First, we evaluate the three variants of the algorithm to decide which one is the most promising. Second, we compare the best MODEBI variant with GDE3 and with the Bayesian Optimization method MACE [35]. Finally, we analyze the impact of randomness over the MODEBI algorithm.

4.3.1 Results of MODEBI

We performed three independent experiments to evaluate the performance of the MODEBI algorithm for optimizing two low-dropout voltage regulators. The experiments included three variants of the proposed algorithm. Because using a Gaussian process model as a surrogate for the circuit simulator should allow us to obtain feasible solutions faster, the variants were aimed at determining the combination of offspring selection methods and population survival approaches that produce better results.

4.4 Strengths and Limitations of MODEBI

4.4.1 Conclusions

In this chapter, we proposed an innovative multi-objective optimization algorithm for automated circuit sizing, called MODEBI. It is designed to handle circuits with a high number of design variables (order of tens), several PVT corners, and many (10+) conflicting responses that must meet the specifications. In this context, MODEBI is a design optimization method that uses an Evolutionary Algorithm inspired from GDE3 to explore the complex hyperspace of the design variables and the PVT corners. Additionally, it employs Gaussian processes as a surrogate for expensive circuit simulations, effectively boosting the convergence of the Evolutionary Algorithm.

Currently, most state-of-the-art algorithms solve multi-objective tasks by pruning them to single-objective optimization tasks or including a priori bias. However, the proposed method employs Pareto dominance, based on which it directly explores over the multi-objective space. Thus, it can provide a population of various solutions to circuit designers.

Chapter 5

The Evolutionary Bayesian Optimization (EBO)

In this chapter, we introduce a novel strategy for automated circuit sizing called Evolutionary Bayesian Optimization (EBO). Our framework integrates the strengths of two leading approaches from the literature. On one hand, it adopts the ML-enhanced EAs approach to global search through evolutionary computation, which has been demonstrated to be effective in both earlier studies [19] and recent advancements (Chapter 4). On the other hand, similar to Bayesian Optimization, our method fully leverages the powerful ML model by actively searching for its optima rather than merely using it as a preselection tool. Additionally, EBO performs true multi-objective optimization and employs population-based search, enabling it to fully utilize parallel simulation infrastructure.

The framework consists of two nested searching mechanisms. The inner one, called virtual evolution, searches for the optima of the ML model. The outer one uses the real circuit simulator to confirm the quality of the solutions produced by the inner search. In addition, it updates the ML model using the new data points available. This process is repeated until the simulation budget is spent.

To show the capabilities of EBO we introduce two particular algorithms developed in our framework. They both use Differential Evolution offspring generation engines and Gaussian Processes, but they differ in terms of survival policies.

The main contributions of this Chapter are the following: (I) We propose a novel framework for automated circuit sizing; (II) We develop two algorithms following the guidelines of this framework; (III) We analyze the performance of the algorithms on two complex proprietary circuits to provide guidance for forthcoming algorithm developments, to measure the impact of the hyperparameters and randomness, and to show that our strategy achieves better results than the state-of-the-art methods. (IV) We create two case studies targeting circuits from the literature to show the practicality of the framework.

5.1 Framework

EBO represents a combination between an evolutionary framework and Bayesian Optimization. The key concept of this framework is the integration of the virtual evolution. EBO algorithms are using two types of optimization steps, real ones and virtual ones. In the virtual steps, the solutions are evaluated on the ML model, which is used as a surrogate for the circuit simulator. When a certain condition is met, the algorithm performs a real step of optimization. Specifically, it evaluates the last population of the virtual evolution using the real circuit simulator. The real steps are performed to ensure the ML model is still accurate, and to verify the quality of the solutions. Thanks to its intrinsic philosophy of population-based evolution, EBO takes full advantage of parallel simulation infrastructure.

The flow of a generic EBO algorithm is presented in Figure 5.1. First, the initial population is randomly generated and evaluated. Second, the algorithm performs virtual evolution until a certain condition is met. Then, the algorithms performs a step of real evolution. Every time the real simulator is used, the surrogate ML model is retrained using the newly available data points.

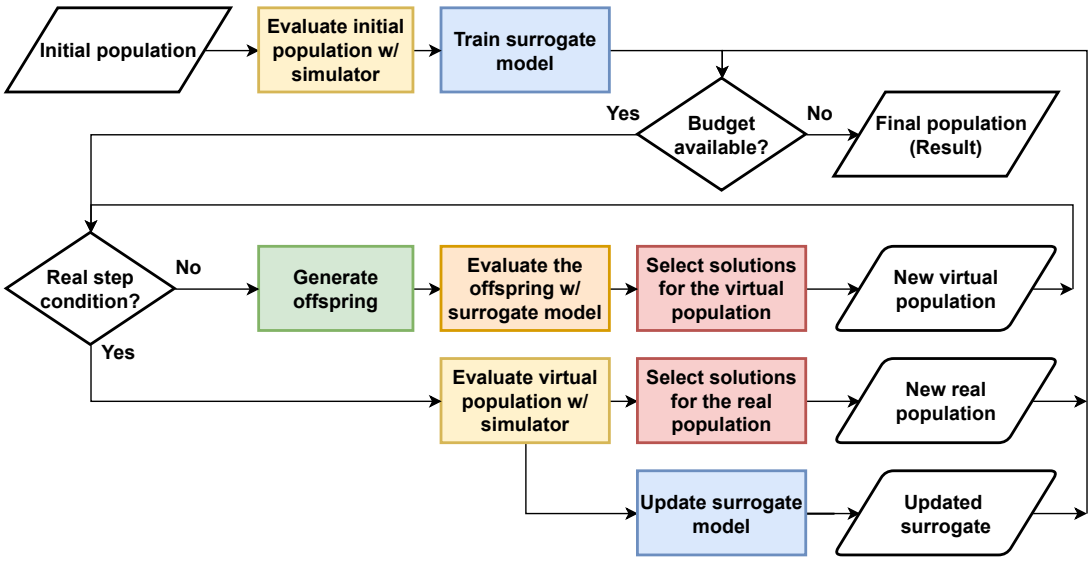

Fig. 5.1 Flow diagram of a generic EBO algorithm

5.2 EBO Algorithms

We propose two particular EBO algorithms, which differ in terms of the survival policies employed. The algorithms use the most common Differential Evolution operator DE/rand/1/bin [46] as the recombination engine. In terms of ML models, the algorithms use separate GPs for each circuit response.

5.3 EBO Evaluation

This section is structured as follows: We start by providing the premises for the evaluation. Then we present a general comparison between the two proposed Evolutionary Bayesian Optimization (EBO) algorithms. Next we go through the tuning process of the virtual evolution's steps number. Afterwards, we make a comparison between our method and state-of-the-art algorithms and a timing assessment. Finally, we assess the robustness of the proposed EBO algorithms.

5.4 Strengths and Limitations of EBO

In the scope of this chapter, we propose Evolutionary Bayesian Optimization (EBO), a new strategy for automated circuit sizing. The framework is a synergy of the two most promising approaches from existing literature, effectively harnessing the best aspects of each. From one perspective it follows the ML enhanced evolutionary algorithms' strategy of global search using evolutionary computation. From another perspective, our technique fully takes advantage of the powerful ML model by searching for its optima, just as Bayesian Optimization. In summary, the EBO framework employs a sequence of real and virtual evolution steps in an alternating fashion to identify the problem's optima.

The EBO framework has some intrinsic benefits. First, it is truly multi-objective, eliminating the drawbacks associated with constructing figure of merit functions. Second, it inherently uses large batches of simulations, taking full advantage of parallel simulation infrastructure. Third, being a diversity driven population based method, it focuses not only on finding a good solution, but on describing the entire Pareto front. Thus, it gives the designers access to various trade-offs. Fourth, the algorithms developed in the EBO framework do not use the uncertainty measures of the ML models. This not only eliminates the risk associated with large variance, but also makes the framework compatible with ML models that do not provide uncertainty measures.

Chapter 6

Operating Corners' Management

This chapter is based on the following articles: [51], [52], [53], [54]

Operating corners are obtained by combining the relevant values of the operating conditions. Managing corners simulations wisely can lead to important budget reduction. Furthermore, it can make the difference between meeting and not meeting the constraints.

One naive but common approach in the literature is to optimize for the nominal corner and check the other corners at the end of the optimization. This minimizes the number of circuit simulations but results in an algorithm that lacks robustness. With this method, the circuit might fail to meet specifications across all corners. Another approach is to identify the worst-case corner at the start and optimize for it. However, due to varying interactions between circuit elements under different operating conditions, the worst-case corner may shift during optimization. The safest approach, as demonstrated in previous chapters, is to evaluate all corners continuously. However, this consumes most of the simulation budget on corners that are often irrelevant.

In this chapter we present several corner management techniques: First, we discuss the two-steps sizing approach in Section 6.1. Then, in Section 6.2 we introduce the Periodic Worst-Corners Selection mechanism. In Section 6.3 we combine the two methods to build an efficiency centric circuit sizer. Finally, we discuss a different paradigm of corners management in 6.4.

6.1 Two-steps Circuit Sizing

An efficient ML model enhanced evolutionary algorithm based optimizer is MODEBI, presented in Chapter 4. It uses different metrics for solutions that meet the specifications (feasible) and those that do not, leading to a true multi-objective optimization approach after feasible solutions are found. Furthermore, the MODEBI algorithm employs separate models for each circuit response, resulting in more accurate predictions.

Optimizing for all corners, as done by the MODEBI algorithm, is safe. It guaranties the optimization takes into account the real worst-case scenario. Nonetheless, optimizing

for the worst corner is not the most efficient strategy. Even more important, it requires numerous simulations.

Circuit designers take a different approach when sizing manually. They start by optimizing for the nominal corner in order to identify promising sizing candidates. After that, they begin to consider extreme operating conditions. We propose an automated circuit sizing method that mimics this designer-like approach. As a result, fewer circuit simulations are spent on unpromising candidates during the early stages of optimization, which is especially important for highly unstable circuits.

6.1.1 The "designer-like" approach

The "designer-like" optimization is a simple technique inspired by the way analog designers approach circuit dimensioning. The simplicity of this method lies in its two-step process. In the first step, the circuit is optimized at the nominal operating corner, allowing the algorithm to converge to a stable area with a low number of simulations. Once a specific condition is met, the algorithm enters the second optimization phase and begins to consider the extreme operating corners.

6.2 Periodic Worst-Corner Selection (PWCS) Mechanism

Reducing the number of design-verification loops is crucial to assuring a good time to market for new circuits. Typically, designers select the most unfavorable PVT corner in advance, leveraging their expertise. Going further, they perform circuit sizing for this condition and, in the end, the design is verified in all PVT corners to demonstrate its robustness. This minimizes the number of circuit simulations required for circuit sizing, but often leads to multiple design-verification loops because the solution which was feasible for the PVT corner chosen up-front turns out not to feasible in all other corners.

In a designer-like approach adapted for automated circuit sizing requires a second step of PVT aware optimization, as presented in Section 6.1. However, this is computationally expensive. In this context, we propose integrating PVT verification in the circuit sizing phase in a more intelligent manner by periodically selecting the worst PVT corners and performing the sizing only in those. This allows obtaining similar results as if optimizing in all corners while respecting the simulation budget.

6.2.1 Method

In MODEBI we proposed evaluating all the corners of all solutions at each iteration in an effort to minimize the number of design-verification loops that take place before an adequate solution is found. Usually in circuit sizing tasks, finding feasible solutions in

some corners can take significantly more simulations than in other corners. In essence, the optimization process comes to a point where most solutions in the population are already feasible in one or more PVT corners, but it is struggling to find feasible solutions in other, more difficult ones.

We propose the MODEBI-CM algorithm, an improvement over the MODEBI algorithm which uses a *worst corner selection* mechanism so that simulations are not wasted on easy corners which are likely within specifications already if the *worst corners* meet them.

6.3 Efficiency Centric Sizer

While the aforementioned techniques (Sections 6.1 and 6.2) have each demonstrated effectiveness on their own, to our knowledge, no attempts have been made to combine various speed-up strategies into a singular, comprehensive sizer. In this section, we introduce an efficiency-centric automatic circuit sizer that consolidates multiple approaches within a single algorithm.

6.3.1 Method

The efficiency-centric automatic circuit sizer is a versatile framework applicable to various evolutionary methods. It extends beyond traditional evolutionary algorithms (EAs) to include ML-enhanced EAs. This method streamlines the sizing process across three levels: initially, employing Latin Hypercube Sampling (LHS) [55] for initial population selection to mitigate randomness and decrease the average budget required for the optimization [56]; beginning the optimization with the nominal operating corner to obtain a "warm start" (Section 6.1) at minimal simulation cost; and then optimizing the circuit on a periodically updated subset of the most challenging operating corners (Section 6.2), validating the most promising solutions against the full corner set.

Algorithm 2 outlines the workflow of the efficiency-centric automatic circuit sizer. While many steps are straightforward, some require further explanation. The "archive" functions as a repository that automatically filters out dominated solutions, where a solution is considered dominated if another exists with superior performance across all objectives. The pruning process is performed according to the same domination criterion. To differentiate among multiple nondominated solutions, Crowding Distance [30] is employed to identify the most diverse solution set. The optimization step can proceed for a predetermined number of epochs or until reaching a specified number of feasible solutions. For the worst corner selection step, users have the flexibility to specify the number of challenging corners they wish to focus on.

Algorithm 2: Efficiency-centric automatic circuit sizer

```
solutions = LHS.generate(pop_size);
solutions = EA.optimize(solutions, nominal_corner);
archive.initialize();
while True do
    elite_pop = prune(solutions, elite_size);
    evaluate(elite_pop, all_corners);
    archive.add(elite_pop);
    if maximum budget reached then
        break;
    end if
    worst_corners = select(all_corners, elite_pop);
    solutions = prune(solutions+archive, pop_size);
    solutions = EA.optimize(solutions, worst_corners);
end while
return archive
```

6.4 Simulation Scheduler

While the techniques previously presented in this chapter have each demonstrated effectiveness, they share a significant drawback. They treat all the solutions in a population equally, regardless of their quality. Ideally, we want to spend circuit simulations on promising solutions, to validate their quality, while swiftly discarding low-quality solutions to save simulations.

In this context, we introduce an alternative mechanism for managing corners, capable of reducing the total number of simulations required by sizing algorithms by more than 70%. This method employs a prioritized list of corners, ordered by their difficulty level. Depending on the use-case, the list might include the nominal corner or not. Solutions undergo sequential evaluation across these corners, with the assessment of each solution halting as soon as it fails to meet the specifications for a given corner. Consequently, this approach minimizes the simulations expended on lower-quality solutions, concentrating the optimization efforts on the more challenging corners.

6.4.1 Method

The foundation of our corner management technique is a simulation scheduler. For optimal operation, it is essential to arrange the list of corners in descending order based on their difficulty level. The proposed method uses the Constraint Violation (CV) [57] metric to rank the operating corners based on their difficulty. CV represents the average of the normalized deviations from the specification of a particular solution in a certain

operating corner (Eq. 6.1).

$$\mathrm{CV}(x) = \frac{1}{N_{\mathrm{constr}}} \sum_{i=1}^{N_{\mathrm{constr}}} \frac{\mathrm{deviation}_i(x)}{\mathrm{reference}_i} \tag{6.1}$$

For a given set of solutions, the average CV for each corner is computed. The corners with the highest CV values are then considered the worst-case corners and ranked first. As opposed to the Periodic Worst-Corner Selection mechanism (Section 6.2) the proposed method performs this ranking just once, at the start of the optimization process, using the randomly generated initial set of proposed solutions.

The fundamental operation of our scheduling method is structured in the following manner: Once the designated evolutionary algorithm produces a fresh batch (population) of potential solutions, these are first assessed in the toughest corner as ranked on our difficulty list. Any solution that fails to meet the specifications for this corner is allocated a specific metric value. Only those solutions that satisfy the requirements in the most challenging corner proceed to be evaluated in the next corner from the difficulty list. The process continues until each solution has either been assigned a metric value or has been evaluated across all corners on the list. If a solution meets the specifications for every corner, it receives a metric value of zero.

The metric implemented in the proposed scheduling method is computed based on the number of corners evaluated for a specific solution (Equation 6.2).

$$\mathrm{Metric}(x) = N_{\mathrm{corners}} - N_{\mathrm{passed}}(x) + \mathrm{CV}(x, N_{\mathrm{passed}} + 1) \tag{6.2}$$

In particular, this metric is obtained by subtracting the number of corners successfully passed by the solution from the overall number of corners. Following this, the CV (Equation 6.1) is computed for the corner that was evaluated but not passed by the solution. This CV is then incorporated into the metric to distinguish between solutions that manage to pass an identical number of corners.

Chapter 7

Conclusions

Automated circuit sizing is a trending topic due to its potential to reduce semiconductor product time-to-market while improving employee satisfaction. It seeks to replace the experience-driven iterative process of manual design with multi-objective optimization algorithms. The most popular techniques for automated circuit sizing are hybrid methods combining ML models with metaheuristic search engines like evolutionary algorithms. Specifically, the most advanced methods rely on ML-enhanced evolutionary algorithms and Bayesian optimization. Recent efforts have focused on improving these methods by introducing novel techniques that reduce the simulation budget, such as corner management strategies.

In this thesis, we aim to advance the state-of-the-art by presenting four key studies. First, in Chapter 3, we analyze five state-of-the-art evolutionary algorithms to identify the most effective one for circuit sizing tasks. Second, in Chapter 4, we propose a new ML-enhanced evolutionary algorithm that employs Gaussian Processes to preselect the most promising offspring generated by a Differential Evolution-based engine. Third, in Chapter 5, we introduce a novel framework for automated circuit sizing, combining the strengths of ML-enhanced evolutionary algorithms and Bayesian optimization. Finally, in Chapter 6, we address the challenge of managing operating corners to improve the efficiency of circuit sizing methods. To demonstrate the performance of our proposed solutions, we compare them against state-of-the-art algorithms using five circuits, including three proprietary designs and two open-source ones. Throughout the research, our focus was not only on performance but also on the practicality and robustness of our methods.

The results presented in Chapter 3 demonstrate that evolutionary algorithms (EAs) are an effective and reliable foundation for circuit sizing tasks. We evaluated five state-of-the-art, biology-inspired [58] EAs: Non-dominated Sorting Genetic Algorithm II (NSGAII) [30], Non-dominated Sorting Genetic Algorithm III (NSGAIII) [59], General Differential Evolution 3 (GDE3) [46], Indicator Based Evolutionary Algorithm (IBEA) [47], and Strength Pareto Evolutionary Algorithm 2 (SPEA2) [60]. Our assessment considered their performance, diversity preservation, and versatility. The findings highlight that

GDE3 consistently performs well across all scenarios and excels in preserving solution diversity, particularly due to its potential of performing well with a large population. As a result, we conclude that GDE3 is the most promising EA for automated circuit sizing. While EAs are highly effective for sampling the design parameter space, relying solely on them leads to a high number of simulations. Thus, our goal is to develop a more efficient circuit sizing algorithm by integrating GDE3 with a surrogate model.

In Chapter 4, we introduced a novel multi-objective optimization algorithm for automated circuit sizing, called Multi-objective Optimization based on Differential Evolution and Bayesian Inference (MODEBI). This method uses an EA inspired by GDE3 to explore the complex hyperspace of design variables and operating corners. Additionally, it employs Gaussian processes as a surrogate model to reduce the need for costly circuit simulations, thereby significantly enhancing convergence. MODEBI is specifically designed to handle circuits with many design variables, multiple operating corners, and numerous conflicting responses that must meet specifications. Unlike most state-of-the-art algorithms, which convert multi-objective tasks into single-objective ones or introduce a priori bias, MODEBI utilizes Pareto dominance to directly explore the multi-objective space. The results demonstrate its superior performance compared to both GDE3 and the state-of-the-art Bayesian optimization method, MACE [35]. Although MODEBI is a powerful algorithm, it has certain limitations. First, it tends to be elitist, making it less versatile and robust. Second, its performance is quite sensitive to hyperparameter tuning, which impacts its practicality.

To overcome the limitations of MODEBI and create an algorithm that is both efficient and highly robust, we developed the Evolutionary Bayesian Optimization (EBO) framework in Chapter 5. This framework synergizes the two most promising approaches from the literature, effectively combining their strengths. From one perspective, it adopts the global search strategy of ML-enhanced evolutionary algorithms by using evolutionary computation. From another, it leverages the power of ML models by searching for their optima, similar to Bayesian Optimization. The EBO framework is based on the key concept of "virtual evolution". It alternates between real and virtual evolution steps to identify optimal solutions. Like MODEBI, it employs Pareto dominance for multi-objective optimization and utilizes large batches of simulations. On the other hand, unlike MODEBI, EBO stays true to the intrinsic philosophy of the EA, without altering its workflow. This approach minimizes the number of hyperparameters, while ensuring diversity preservation and robustness if paired with the appropriate EA. Results from four circuits demonstrate EBO's superiority over MODEBI, GDE3, and MACE. While developing new optimization algorithms for circuit sizing is exciting, we acknowledge the law of diminishing returns and shift our focus to enhancing performance through strategies like operating corners management.

In Chapter 6, we propose several performance enhancement methods under the umbrella of 'operating corners management.' Operating corners are derived by combining

relevant operating condition values. For robust optimization, each circuit configuration must be validated across multiple operating corners. Efficient management of these simulations can significantly reduce the simulation budget. The safe approach used in MODEBI and EBO validates all circuit configurations in every operating corner. However, this approach quickly consumes the simulation budget.

One basic corner management technique (Section 6.1) optimizes for the nominal corner until the feasibility region is found, then start checking all the corners for the rest of the optimization. This approach provides a 'warm-start' for corner-aware optimization with minimal simulation cost. An alternative method is the periodic worst-corner selection mechanism (Section 6.2), which maintains algorithm robustness while significantly reducing the simulation budget. Additionally, combining these two methods, as shown in Section 6.3, maximizes efficiency.

Although these methods are effective, they have a key drawback: they treat all circuit configurations equally, regardless of quality. Ideally, simulations should be spent on validating promising solutions, while quickly discarding low-quality ones to conserve resources. The simulation scheduler described in Section 6.4 sequentially evaluates circuit configurations across corners, halting the evaluation of a particular configuration as soon as it doesn't meet the specifications in one of the operating corners.

The key advantage of the proposed corner management techniques is their flexibility—they are not tied to any specific sizing algorithm. They function as general performance enhancement methods for any evolutionary computation-based circuit sizing approach.

In summary, this research advances the state-of-the-art on multiple levels. We evaluate the performance of evolutionary algorithms in circuit sizing tasks to identify the most promising approach. Building on this, we propose two novel circuit sizing methods that address limitations in existing techniques. Finally, we develop several performance enhancement strategies to further boost the efficiency of our proposed methods.

7.1 Original contributions

The contributions related to the analysis of state-of-the-art evolutionary algorithms presented in Chapter 3 are as follows:

- Hyperparameter tuning on synthetic benchmarks [43, 44]. This task aims to determine the best configurations for state-of-the-art evolutionary algorithms using computationally inexpensive problems that resemble circuit sizing tasks. The results provide a valuable resource for any research focused on automated circuit sizing with evolutionary computation.

- In-depth analysis of state-of-the-art evolutionary algorithms on proprietary cutting-edge analog circuits [43, 44]. This work offers insight into how specific character-

istics of evolutionary algorithms influence their suitability for circuit sizing. The analysis considers not only performance but also the versatility and robustness of the algorithms.

The main contributions related to Machine Learning-enhanced evolutionary algorithms presented in Chapter 4 are the following:

- A novel multi-objective optimization method combining a Differential Evolution-based algorithm with Gaussian Processes, designed to excel in problems with numerous variables, objectives, and constraints [48]. The algorithm achieves a speedup of over three times compared to classic evolutionary algorithms and performs better than state-of-the-art circuit sizing methods.
- Several innovative population survival policies and offspring selection algorithms that minimize the number of real simulations required to complete the optimization process while preserving solution diversity [48, 49].

The contributions regarding the application of Bayesian optimization in circuit sizing presented in Chapter 5 are as follows:

- A novel framework for automated circuit sizing that combines ML-enhanced evolutionary algorithms with Bayesian optimization. The proposed method integrates the two approaches, leveraging the strengths of both.
- Two circuit sizing algorithms developed according to this novel framework. The results demonstrate significant improvements in performance and robustness compared to state-of-the-art methods.
- Two case studies involving circuits from the literature to demonstrate the practicality of the framework. While the results on proprietary circuits show that the algorithms are capable of handling cutting-edge sizing problems, some details cannot be disclosed due to proprietary constraints. To offer broader insights, the proposed algorithms are also tested on popular circuits from the literature.

The contributions regarding performance enhancement through operating corners management presented in Chapter 6 are as follows:

- A periodic worst-corner selection mechanism that ensures the robustness of the algorithm while reducing the simulation budget required to achieve optimal solutions [51].
- An efficiency-centric circuit sizer [54] that combines the periodic worst-corner selection mechanism with a two-step sizing technique [52]. This technique, inspired by manual circuit sizing, begins with nominal corner sizing followed by operating-corner-aware optimization.

- A novel simulation scheduler [53] that sequentially evaluates circuit configurations across operating corners, reducing the number of simulations allocated to low-quality solutions.

7.2 List of original publications

1. [43] Stănescu, M., Vişan, C., Sandu, G., Cucu, H., Diaconu, C., Buzo, A., & Pelz, G. (2021, October). Multi-objective optimization algorithms for automated circuit sizing of analog/mixed-signal circuits. In 2021 International Semiconductor Conference (CAS) (pp. 117-120). IEEE, ISI WOS:000853482700022

2. [44] Vişan, C., Pascu, O., Stănescu, M., Cucu, H., Diaconu, C., Buzo, A., & Pelz, G. (2021, October). Versatility and population diversity of evolutionary algorithms in automated circuit sizing applications. In 2021 International Conference on Speech Technology and Human-Computer Dialogue (SpeD) (pp. 68-73). IEEE, ISI WOS:000786794700013

3. [61] Pascu, O., Visan, C., Stănescu, M., Cucu, H., Diaconu, C., Buzo, A., & Pelz, G. (2022, October). Efficient Modeling of PVT Variation for Mixed-Signal Circuit Sizing. In 2022 International Semiconductor Conference (CAS) (pp. 105-108). IEEE.

4. [48] Vişan, C., Pascu, O., Stănescu, M., Şandru, E. D., Diaconu, C., Buzo, A., Pelz, G. & Cucu, H. (2022). Automated circuit sizing with multi-objective optimization based on differential evolution and Bayesian inference. Knowledge-Based Systems, 258, 109987, ISI WOS:000883752300005

5. [49] Vișan, C., Pascu, O., Stănescu, M., Cucu, H., Diaconu C., Buzo, A. & Pelz, G., (2022). IMPROVING MODEBI: MULTI-OBJECTIVE OPTIMIZATION BASED ON DIFFERENTIAL EVOLUTION AND BAYESIAN INFERENCE. UPB Scientific Bulletin, Series C: Electrical Engineering and Computer Science, Vol. 84, Issue 4, Pg. 181-196, ISSN 2286-3540, ISI WOS:000907279800011

6. [51] Pascu, O., Visan, C., Nicolae, G., Boldeanu, M., Cucu, H., Diaconu, C., ... & Pelz, G. (2023, August). Efficient Multi-Objective Optimization for PVT Variation-Aware Circuit Sizing Using Surrogate Models and Smart Corner Sampling. In 2023 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED) (pp. 1-6). IEEE, ISI WOS:001073659300049

7. [52] Vişan, C., Sieberer, M., & Cucu, H. (2023, October). Designer-like Automated Circuit Sizing for Multiloop LDO. In 2023 International Semiconductor Conference (CAS) (pp. 103-106). IEEE.

8. [56] Nicolae, G., Visan, C., Curavale, D., Boldeanu, M., Cucu, H., Buzo, A., & Pelz, G. (2023, October). A Study on Initial Population Sampling for Multi-Objective Optimization based on Differential Evolution and Bayesian Inference. In 2023 International Conference on Speech Technology and Human-Computer Dialogue (SpeD) (pp. 128-132). IEEE.

9. [53] Visan, C., Curavale, D., Nicolae, G., Boldeanu, M., Cucu, H. & Buzo, A. (2024, June). A novel simulations scheduler for automated circuit sizing algorithms. In 2024 International Conference on Synthesis, Modeling, Analysis and Simulation Methods, and Applications to Circuit Design (SMACD)

10. [54] Visan, C., Pleșa, C.S. & Cucu, H. (2024, November). Efficiency-centric Automatic Circuit Design for Feed-forward Operational Amplifier. (ICECS), *accepted for publication*

7.3 Perspectives for further developments

The methods proposed in this thesis are well-defined. Only a few specific aspects require further development. The most significant is finding an adaptive mechanism for selecting the appropriate number of virtual steps in the Evolutionary Bayesian Optimization (Chapter 5). Currently, we empirically set the number of virtual steps per real step between 30 and 80, depending on the problem. To prevent potential issues, the mechanism could be designed to select within this range. One early idea is to halt virtual evolution if the offspring deviate too far from any solution evaluated by the real circuit simulator. This prevents reliance on the ML model in regions where it may be unreliable.

Another method requiring further development is the simulation scheduler described in Section 6.4. Since it evaluates circuit configurations sequentially across operating corners, it cannot fully exploit highly parallel infrastructure. We aim to develop a system for organizing simulation batches that maximizes parallelism. A straightforward solution is to fill incomplete batches with simulations from future corners, as these are likely to be needed. Additionally, the simulation scheduler has been applied only in the context of classic evolutionary algorithms. It would be valuable to explore this mechanism with more advanced circuit sizing methods, such as ML-enhanced evolutionary algorithms (Chapter 4) and Bayesian Optimization techniques (Chapter 5).

Initial population sampling is an important aspect worth discussing, as it influences all the methods introduced in this thesis. So far, we have primarily relied on random sampling, a standard approach in evolutionary algorithms. However, more sophisticated techniques, such as Latin Hypercube Sampling [55] or Sobol sampling [62], could provide more relevant data, as suggested in [56]. We employed Latin Hypercube Sampling in Section 6.3, but did not specifically analyze its performance compared to random sampling.

Shifting from specific improvements to broader avenues of exploration, we begin with online circuit modeling. Gaussian Processes are powerful models that achieve impressive accuracy with small datasets. However, they do not scale well for large datasets, with a time complexity of $O(N^3)$ for training and $O(N^2)$ for prediction. In circuit sizing, the simulation budget often reaches tens of thousands, while Gaussian Processes effectively handle around five to ten thousand. One area to explore is the development of training point selection mechanisms for datasets larger than ten thousand simulations. For instance, clustering techniques could be used to either discard some of densely packed training points or to focus model training on relevant areas of the hyperspace.

Alternatively, other Machine Learning models can be integrated into circuit sizing algorithms. For instance, we experimented with Random Forests [63], and early results indicate a substantial reduction in the algorithms' internal delay. However, Random Forest models are less accurate than Gaussian Processes, which can sometimes impact convergence. Another avenue worth exploring is artificial Neural Networks. We empirically found that Neural Networks are not sufficiently accurate with training sets smaller than ten thousand simulations. However, given the advances in Neural Network models, further investigation is required. A compromise could be to switch between surrogate models during optimization. The algorithm could use Gaussian Processes for the first ten thousand simulations, then transition to Neural Networks.

Another application of Neural Networks in circuit sizing is encoding the real input space into a latent space. Specifically, the original input data can be mapped to a continuous, lower-dimensional feature space where Gaussian Processes can be applied. This is achievable through the use of Variational Autoencoders [64].

An alternative to the MODEBI and EBO algorithms could incorporate local refinements using gradient search. Similar to memetic algorithms, this approach leverages the differentiable surrogate model to solve the sizing problem. Replacing some evolutionary computation-based virtual steps with gradient steps may result in faster convergence toward the model's optima.

Recently, Reinforcement Learning and Transfer Learning have gained popularity in the automated circuit design community. One notable method combines Graph Convolutional Neural Networks with Reinforcement Learning, first introduced in [65] and further developed in [66]. At first glance, this approach offers a structural advantage over evolutionary computation and Bayesian optimization methods. By modeling the circuit as a graph, a deep reinforcement learning agent trained on one circuit can be applied to other topologies. With sufficient training, a Reinforcement Learning agent could outperform human designers and automated circuit sizing methods that do not leverage Transfer Learning.

References

[1] Cristian Manolache, Alexandru Caranica, Marius Stănescu, Horia Cucu, Andi Buzo, Cristian Diaconu, and Georg Pelz. Advanced operating conditions search applied in analog circuit verification. In *2022 18th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*, pages 1–4. IEEE, 2022.

[2] Alecsandra Rusu, Emilian David, Marina Țopa, Vasile Grosu, Andi Buzo, and Georg Pelz. Improvement and performance evaluation of an adaptive method for integrated circuits pre-silicon verification. In *2023 International Symposium on Signals, Circuits and Systems (ISSCS)*, pages 1–4. IEEE, 2023.

[3] António Gusmão, Fábio Passos, Ricardo Póvoa, Nuno Horta, Nuno Lourenço, and Ricardo Martins. Semi-supervised artificial neural networks towards analog ic placement recommender. In *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5. IEEE, 2020.

[4] B Carbunescu-Stoenescu, E David, M Topa, A Buzo, and G Pelz. Distribution modelling for yield analysis using variational autoencoders. In *2023 International Symposium on Signals, Circuits and Systems (ISSCS)*, pages 1–5. IEEE, 2023.

[5] Vasile Grosu, Liviu Goras, Emilian David, and Georg Pelz. Using neural network based active learning for modelling integrated circuits behavior. In *2023 International Symposium on Signals, Circuits and Systems (ISSCS)*, pages 1–4. IEEE, 2023.

[6] Vasile Grosu, Emilian David, Liviu Goras, and Georg Pelz. Modelling integrated circuits behavior using an active learning approach based on gaussian process regression. In *2023 International Semiconductor Conference (CAS)*, pages 245–248. IEEE, 2023.

[7] Nuno Horta. Analogue and mixed-signal systems topologies exploration using symbolic methods. *Analog Integrated Circuits and Signal Processing*, 31:161–176, 2002.

[8] Walter Daems, Georges Gielen, and Willy Sansen. Simulation-based generation of posynomial performance models for the sizing of analog integrated circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 22(5):517–534, 2003.

[9] Stephen P. Boyd and Seung Jean Kim. Geometric programming for circuit optimization. In *Proceedings of the 2005 International Symposium on Physical Design*, ISPD '05, page 44–46, New York, NY, USA, 2005. Association for Computing Machinery.

[10] Abdelrahman Sayed, Ahmed Nader Mohieldin, and Mohsen Mahroos. A fast and accurate geometric programming technique for analog circuits sizing. In *2019 31st International Conference on Microelectronics (ICM)*, pages 316–319, 2019.

[11] Ye Wang, Michael Orshansky, and Constantine Caramanis. Enabling efficient analog synthesis by coupling sparse regression and polynomial optimization. In *Proceedings of the 51st Annual Design Automation Conference*, pages 1–6, 2014.

[12] W Lyu, P Xue, F Yang, C Yan, Z Hong, X Zeng, and D Zhou. An efficient bayesian optimization approach for automated optimization of analog circuits. *IEEE Trans. Circuits Syst. I Regul. Pap.*, 65(6):1954–1967, June 2018.

[13] Bo Liu, Yan Wang, Zhiping Yu, Leibo Liu, Miao Li, Zheng Wang, Jing Lu, and Francisco V Fernández. Analog circuit optimization system based on hybrid evolutionary algorithms. *Integration*, 42(2):137–148, 2009.

[14] R. Phelps, M. Krasnicki, R.A. Rutenbar, L.R. Carley, and J.R. Hellums. 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, 2000.

[15] Jacopo Panerati, Donatella Sciuto, Giovanni Beltrame, et al. Optimization strategies in design space exploration. In *Handbook of Hardware/Software Codesign*, pages 189–216. Springer Netherlands, 2017.

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

[17] Vipul Kumar Mishra and Anirban Sengupta. Mo-pse: Adaptive multi-objective particle swarm optimization based design space exploration in architectural synthesis for application specific processor design. *Advances in Engineering Software*, 67:111–124, 2014.

[18] B Liu, N Deferm, D Zhao, P Reynaert, and G G E Gielen. An efficient High-Frequency linear RF amplifier synthesis method based on evolutionary computation and machine learning techniques. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, 31(7):981–993, July 2012.

[19] Bo Liu, Dixian Zhao, Patrick Reynaert, and Georges G. E. Gielen. Gaspad: A general and efficient mm-wave integrated circuit synthesis method based on surrogate model assisted evolutionary algorithm. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 33(2):169–182, 2014.

[20] Ahmet Faruk Budak, Miguel Gandara, Wei Shi, David Z Pan, Nan Sun, and Bo Liu. 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, 2021.

[21] Sen Yin, Ruitao Wang, Jian Zhang, Xiaosen Liu, and Yan Wang. Fast surrogate-assisted constrained multi-objective optimization for analog circuit sizing via self-adaptive incremental learning. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2022.

[22] Bo Liu, Hadi Aliakbarian, Soheil Radiom, Guy A E Vandenbosch, and Georges Gielen. Efficient multi-objective synthesis for microwave components based on computational intelligence techniques. In *Proceedings of the 49th Annual Design Automation Conference*, DAC '12, pages 542–548, New York, NY, USA, 2012. ACM.

[23] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

[24] Matthias Seeger. Gaussian processes for machine learning. *International journal of neural systems*, 14(02):69–106, 2004.

[25] Rainer Storn and Kenneth Price. Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341, 1997.

[26] Nuno Lourenço and Nuno Horta. GENOM-POF: multi-objective evolutionary synthesis of analog ICs with corners validation. In *Proceedings of the 14th annual conference on Genetic and evolutionary computation*, GECCO '12, pages 1119–1126, July 2012.

[27] Nuno Lourenço, Ricardo Martins, António Canelas, Ricardo Póvoa, and Nuno Horta. AIDA: Layout-aware analog circuit-level sizing with in-loop layout generation. *Integration*, 55:316–329, September 2016.

[28] A Canelas, R Martins, R Povoa, N Lourenco, and N Horta. Efficient yield optimization method using a variable K-Means algorithm for analog IC sizing. In *Design, Automation Test in Europe Conference Exhibition (DATE), 2017*, pages 1201–1206, March 2017.

[29] Kalyanmoy Deb. An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*, 186(2):311–338, 2000.

[30] K Deb, A Pratap, S Agarwal, and T Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.*, 6(2):182–197, April 2002.

[31] Manuel Barros, Jorge Guilherme, and Nuno Horta. Ga-svm feasibility model and optimization kernel applied to analog ic design automation. In *Proceedings of the 17th ACM Great Lakes symposium on VLSI*, pages 469–472, 2007.

[32] Bo Liu, Francisco V Fernández, Qingfu Zhang, Murat Pak, Suha Sipahi, and Georges Gielen. An enhanced moea/d-de and its application to multiobjective analog cell sizing. In *IEEE Congress on Evolutionary Computation*, pages 1–7. IEEE, 2010.

[33] Nuno Lourenço, Ricardo Martins, Manuel Barros, and Nuno Horta. Analog circuit design based on robust pofs using an enhanced moea with svm models. *Analog/RF and Mixed-Signal Circuit Systematic Design*, pages 149–167, 2013.

[34] Eric Brochu, Vlad M. Cora, and Nando de Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *ArXiv*, abs/1012.2599, 2010.

[35] Wenlong Lyu, Fan Yang, Changhao Yan, Dian Zhou, and Xuan Zeng. Batch bayesian optimization via multi-objective acquisition ensemble for automated analog circuit design. In *International Conference on Machine Learning*, pages 3306–3314. proceedings.mlr.press, 2018.

[36] Matthew D. Hoffman, Eric Brochu, and Nando de Freitas. Portfolio allocation for bayesian optimization. *ArXiv*, abs/1009.5419, 2010.

[37] Biao He, Shuhan Zhang, Fan Yang, Changhao Yan, Dian Zhou, and Xuan Zeng. An efficient bayesian optimization approach for analog circuit synthesis via sparse gaussian process modeling. In *2020 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 67–72, 2020.

[38] Ke Li, Renzhi Chen, Guangtao Fu, and Xin Yao. Two-archive evolutionary algorithm for constrained multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 23(2):303–315, 2018.

[39] S Venkatraman and G G Yen. A generic framework for constrained optimization using genetic algorithms. *Trans. Evol. Comp*, 9(4):424–435, August 2005.

[40] Xin Li, Padmini Gopalakrishnan, Yang Xu, and Lawrence T. Pileggi. Robust analog/rf circuit design with projection-based performance modeling. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 26:2–15, 2007.

[41] S K Goudos, K Siakavara, T Samaras, E E Vafiadis, and J N Sahalos. Self-Adaptive differential evolution applied to Real-Valued antenna and microwave design problems. *IEEE Trans. Antennas Propag.*, 59(4):1286–1298, April 2011.

[42] Ivan Voutchkov and Andy Keane. Multi-Objective optimization using surrogates. In Yoel Tenne and Chi-Keong Goh, editors, *Computational Intelligence in Optimization: Applications and Implementations*, pages 155–175. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[43] M Stanescu, C Visan, G Sandu, H Cucu, C Diaconu, A Buzo, and G Pelz. Multi-objective optimization algorithms for automated circuit sizing of analog/ mixed-signal circuits. In *2021 International Semiconductor Conference (CAS)*, pages 1–4. IEEE, October 2021.

[44] C Visan, O Pascu, M Stanescu, H Cucu, C Diaconu, A Buzo, and G Pelz. Versatility and population diversity of evolutionary algorithms in automated circuit sizing applications. In *2021 International Conference on Speech Technology and Human-Computer Dialogue (SpeD)*, pages 1–6. IEEE, October 2021.

[45] Simon Huband, Luigi Barone, Lyndon While, and Phil Hingston. A scalable multi-objective test problem toolkit. In *Evolutionary Multi-Criterion Optimization: Third International Conference, EMO 2005, Guanajuato, Mexico, March 9-11, 2005. Proceedings 3*, pages 280–295. Springer, 2005.

[46] S Kukkonen and J Lampinen. GDE3: the third evolution step of generalized differential evolution. In *2005 IEEE Congress on Evolutionary Computation*, volume 1, pages 443–450 Vol.1, September 2005.

[47] Eckart Zitzler and Simon Künzli. Indicator-Based selection in multiobjective search. In *Parallel Problem Solving from Nature - PPSN VIII*, pages 832–842. Springer Berlin Heidelberg, 2004.

[48] Catalin Visan, Octavian Pascu, Marius Stanescu, Elena-Diana Sandru, Cristian Diaconu, Andi Buzo, Georg Pelz, and Horia Cucu. Automated circuit sizing with multi-objective optimization based on differential evolution and bayesian inference. *Knowledge-Based Systems*, 258:109987, 2022.

[49] Catalin Visan, Octavian Pascu, Marius Stanescu, Horia Cucu, Cristian Diaconu, Andi Buzo, and Georg Pelz. Improving modebi: Multi-objective optimization based on differential evolution and bayesian inference. *UPB Scientific Bulletin*, 84(4), 2022.

[50] Kai Zheng, Ren-Jye Yang, Hongyi Xu, and Jie Hu. A new distribution metric for comparing pareto optimal solutions. *Struct. Multidiscip. Optim.*, 55(1):53–62, January 2017.

[51] Octavian Pascu, Catalin Visan, Georgian Nicolae, Mihai Boldeanu, Horia Cucu, Cristian Diaconu, Andi Buzo, and Georg Pelz. Efficient multi-objective optimization for pvt variation-aware circuit sizing using surrogate models and smart corner sampling. In *2023 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, pages 1–6, 2023.

[52] Cătălin Vişan, Michael Sieberer, and Horia Cucu. Designer-like automated circuit sizing for multiloop ldo. In *2023 International Semiconductor Conference (CAS)*, pages 103–106, 2023.

[53] Catalin Visan, Dan Curavale, Georgian Nicolae, Mihai Boldeanu, Horia Cucu, and Andi Buzo. A novel simulations scheduler for automated circuit sizing algorithms. In *2024 20th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*, pages 1–4. IEEE, 2024.

[54] Catalin Visan, Cosmin-Sorin Plesa, and Horia Cucu. Efficiency-centric automatic circuit design for feed-forward operational amplifier. In *Proceedings of the International Conference on Electronics, Circuits, and Systems (ICECS)*, November 2024. Accepted for publication.

[55] Michael D McKay, Richard J Beckman, and William J Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42(1):55–61, 2000.

[56] Georgian Nicolae, Catalin Visan, Dan Curavale, Mihai Boldeanu, Horia Cucu, Andi Buzo, and Georg Pelz. A study on initial population sampling for multi-objective optimization based on differential evolution and bayesian inference. In *2023 International Conference on Speech Technology and Human-Computer Dialogue (SpeD)*, pages 128–132, 2023.

[57] Sanyou Zeng, Ruwang Jiao, Changhe Li, Xi Li, and Jawdat S Alkasassbeh. A general framework of dynamic constrained multiobjective evolutionary algorithms for constrained optimization. *IEEE transactions on Cybernetics*, 47(9):2678–2688, 2017.

[58] Sinem Akyol and Bilal Alatas. Plant intelligence based metaheuristic optimization algorithms. *Artif Intell*, 47:417–462, 2017.

[59] K Deb and H Jain. An evolutionary Many-Objective optimization algorithm using Reference-Point-Based nondominated sorting approach, part i: Solving problems with box constraints. *IEEE Trans. Evol. Comput.*, 18(4):577–601, August 2014.

[60] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. SPEA2: Improving the strength pareto evolutionary algorithm. *TIK-report*, 103, 2001.

[61] Octavian Pascu, Cătălin Visan, Marius Stănescu, Horia Cucu, Cristian Diaconu, Andi Buzo, and Georg Pelz. Efficient modeling of pvt variation for mixed-signal circuit sizing. In *2022 International Semiconductor Conference (CAS)*, pages 105–108, 2022.

[62] Ilya M. Sobol. On the distribution of points in a cube and the approximate evaluation of integrals. *Ussr Computational Mathematics and Mathematical Physics*, 7:86–112, 1967.

[63] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[64] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2014.

[65] Hanrui Wang, Jiacheng Yang, Hae-Seung Lee, and Song Han. Learning to design circuits. *arXiv preprint arXiv:1812.02734*, 2018.

[66] Hanrui Wang, Kuan Wang, Jiacheng Yang, Linxiao Shen, Nan Sun, Hae-Seung Lee, and Song Han. Gcn-rl circuit designer: Transferable transistor sizing with graph neural networks and reinforcement learning. In *2020 57th ACM/IEEE Design Automation Conference (DAC)*, pages 1–6. IEEE, 2020.