AUTOMATED DESIGN OF CMOS ANALOG INTEGRATED CIRCUITS USING METAHEURISTIC ALGORITHMS

Submitted in partial fulfillment of the requirements
for the award of the degree of
DOCTOR OF PHILOSOPHY

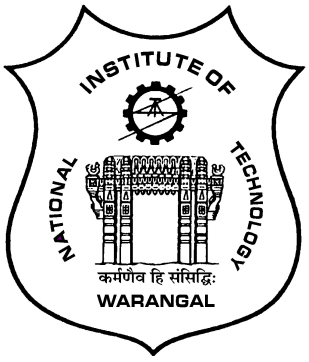
by

M A MUSHAHHID MAJEED

(Roll No: 716029)

Supervisor:

Dr. SREEHARI RAO P
Associate Professor



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL
TELANGANA STATE-506004, INDIA

2020

Dedicated to

GOD Almighty & My family

DECLARATION

This is to certify that the work presented in the thesis entitled "**Automated Design of CMOS Analog Integrated Circuits Using Metaheuristic Algorithms**" is a bonafide work done by me under the supervision of **Dr. P Sreehari Rao**, Professor, Department of Electronics and Communication Engineering, National Institute of Technology Warangal, India and was not submitted elsewhere for the award of any degree.

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea / data / fact / source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

M A Mushahhid Majeed

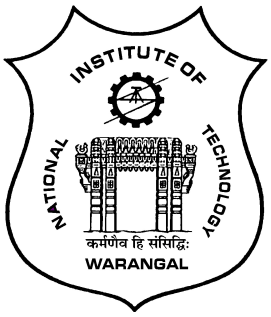
(Roll No: 716029)

Date: .....

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL

TELANGANA STATE-506004, INDIA



CERTIFICATE

This is to certify that the thesis entitled "**Automated Design of CMOS Analog Integrated Circuits Using Metaheuristic Algorithms**", which is being submitted by **Mr. M A Mushahhid Majeed (Roll No: 716029)**, in partial fulfilment for the award of the degree of Doctor of Philosophy to the Department of Electronics and Communication Engineering of National Institute of Technology Warangal, is a record of bonafide research work carried out by him under my supervision and has not been submitted elsewhere for any degree.

Dr. P Sreehari Rao
(Supervisor)
Associate Professor
Department of E.C.E.
N.I.T. Warangal
Warangal - 506004, India

ACKNOWLEDGEMENTS

Many people have devoted immensely in making this accomplishment possible. I take this opportunity to express my heartily obligations to all the people for their moral support, optimism and faith in me throughout this journey.

First and foremost, I would like to express my earnest gratitude to my thesis advisor Dr. P. Sreehari Rao for his consistent guidance, moral support, invaluable advices and being there for me like family. His research enthusiasm inspired me to look into CMOS world and open my own research field.

I would like to thank my D.S.C. committee members; Prof. N. Bheema Rao (Chairman of the D.S.C. committee and Head of the Department), Dr. T V K Hanumanth Rao and Prof. D M Vinod Kumar for their guidance and encouragements. Their timely feedbacks have helped me a lot in shaping my thesis work. I am thankful to all the faculty members and people at the office of our department for helping me directly or indirectly.

I wish to thank all my colleagues and friends at Mixed Signal Design Laboratory for providing a pleasant and friendly working environment. In particular, I must thank Dr. Suresh A., D. Pavan Kumar, Krishna Reddy K, Ravi Chandra, Baba Fayaz D, Shashank, Sri Ramtej, Surekha B, karnakar P, Srikar V, Vasu B, and M.Tech VLSI-SD students (Batch 2017-2018). All of them are admirable people and guileless friends. Their academic knowledge and experience in analog design helped me lot during my earlier times. Their lovely personalities and diverse cultural backgrounds broadened my horizon and enriched my life. I will always remember the time we spent together.

I would like to acknowledge the Ministry of Electronics & Information Technology (MeitY), Government of India, for providing financial assistance for this research work through "Visvesvaraya PhD Scheme for Electronics & IT". I would also like to thank the Department of Information Technology, Ministry of Electronics and Information Technology, Government of India for providing us the resources at Mixed Signal Design Laboratory under Special Manpower Development Program Chip to System for VLSI design and related software (SMDP-C2S).

I am obliged to Dr. Mudasir Bashir, Shashank R, Roshan B, Dr. Hathiram N, Sunil Kumar T and Jailsingh B for being there all the time with me. Their moral support, suggestions and nature of always helping me out in all possible ways has been overwhelming. Finally, I would

like to thank my parents for their understandings and support to pursue my academic goals. Their self-sacrifices, unconditional love and encouragement has always been the source of my strength and I shall be grateful forever. I would like to thank my grandparents for their eternal love and prayers without which I would not be the person I am. Also, I am gratified to my brother, M A Mubashir Majeed, for his support, indulgences and belief in me overtime.

Above all, I thank God for filling me every day with new hopes, strength, purpose and faith.

ABSTRACT

The rapid evolution and widespread of consumer electronics have imposed a significant strain on the Integrated Circuit (IC) industry. Even though, majority of the functionalities are implemented using digital circuitry, some of the inevitable analog tasks in modern IC's, such as conversion, reception and transmission, power regulation and communication with device sensors, lead to the development of mixed signal systems. Unlike digital design, there is a huge requirement of skilled designers to handle the aspects such as performance, non-linearities, tradeoffs, parasitics, etc. Besides continuous downsizing, shorter life cycle of modern electronic systems impose several challenges to meet the performance requirements of analog circuits. However, the low level of abstraction is considered as one of the major factors hindering the development of analog design automation resulting in higher costs and overall time-to-market of an entire IC. Therefore, there is a huge requirement for novel design automation methodologies to aid analog designers in achieving the goals faster and better.

This thesis introduces analog design and automation techniques for circuit sizing of Complementary Metal Oxide Semiconductor (CMOS) analog circuits. Also, this thesis proposes various optimization algorithms, such as, Enhanced Grey Wolf Optimization (EGWO), a hybrid of Whale optimization Algorithm (WOA) and modified Grey Wolf Optimization (mGWO) algorithm and a hybrid of Sine-Cosine Algorithm (SCA) and WOA, to improve the exploration ability of conventional algorithm.

This thesis contributes two different design methodologies, i.e., equation-based and simulation-based techniques, for analog circuit sizing using novel optimization algorithms. Firstly, recently proposed metaheuristic algorithms , such as, Grey Wolf optimization (GWO) algorithm and Whale Optimization Algorithm (WOA), are applied for the design optimization of CMOS analog circuits considering their simplicity and flexibility. In the equation-based methodology, the aspect ratios are calculated by simulating the algorithm in MATLAB that are further used to implement corresponding circuits in cadence analog design environment using 180nm CMOS standard process. However, these algorithm are prone to premature convergence at local optimum solution. Thus, EGWO algorithm is proposed considering the random search agents for position update instead of best search agent to avoid stagnation at local optimum solution by improving the exploration ability. To further increase the exploration, the hybrid of WOA and modified GWO, combining the abilities of two individual algorithms, is also proposed. Here, one of the algorithms concentrate on the exploration of the search space while the other focuses on the exploitation. The proposed algorithms are validated using the set of

classical benchmark functions for convergence and efficiency. To build further credence and to prove their profound existence in the latest state-of-the-art, a statistical study is also conducted over 20 independent runs for evaluating the robustness of the proposed algorithm. The proposed algorithms are applied for circuit sizing problem considering CMOS differential amplifier and two-stage CMOS operational amplifier as test cases with MOS transistor area and power consumption as objectives.

Moreover, a novel simulation-based optimization technique using a SCA-WOA algorithm is also proposed for better accuracy and efficiency of the design. This methodology uses continuous evaluation of Cadence scripts and optimization engine, resulting in the accuracy equal to the models used in the circuit simulator. Similar to other proposed algorithms, the SCA-WOA algorithm is also tested for its efficiency using the above-said classical benchmark functions while performing the statistical study. Two benchmark circuits, i.e., Folded cascode operational transconductance amplifier (FCOTA) and low dropout voltage regulator, were used to validate the proposed simulation-based optimization technique. The circuit sizing of FCOTA is done for three different technologies, i.e., 180 nm, 130 nm and 65 nm. The faster convergence of the proposed algorithms aided in the better performance of the design automation methodology when compared to other competing techniques.

Contents

List of Figures

List of Tables

List of Abbreviations

ABC	**A**rtificial **B**ee **C**olony **A**lgorithm
AC	**A**lternating **C**urrent
ACO	**A**nt **C**olony **O**ptimization
ADC	**A**nalog to **D**igital **C**onverter
AGSAPSO	A hybrid of **A**dvanced **G**ravitational **S**earch **A**lgorithm and **P**article **S**warm **O**ptimization
ALCPSO	A hybrid of **A**ging **L**eader and **C**hallengers algorithm and **P**article **S**warm **O**ptimization
ANN	**A**rtificial **N**eural **N**etwork
AWE	**A**symptotic **W**aveform **E**valuator
BSIM	**B**erkeley **S**hort-Channel **I**nsulated-Gate Field Effect Transistor **M**odel
CAD	**C**omputer **A**ided **D**esign
CBO	**C**olliding **B**odies **O**ptimization
CF	**C**ost **F**unction
CMOS	**C**omplementary **M**etal **O**xide **S**emiconductor
CMRR	**C**ommon **M**ode **R**ejection **R**atio
CO	**C**onvex **O**ptimization
DC	**D**irect **C**urrent
DE	**D**ifferential **E**volution
DIFFAMP	**DIFF**erential **AMP**lifier
EDA	**E**lectronic **D**esign **A**utomation
EGWO	**E**nhanced **G**rey **W**olf **O**ptimization
FCOTA	**F**olded **C**ascode **O**perational **T**ransconductance **A**mplifier
FF	**F**ast **F**ast
FoM	**F**igure-**o**f-**M**erit
FS	**F**ast **S**low
FSOA	**F**ish **S**warm **O**ptimization **A**lgorithm

GA	**G**enetic **A**lgorithm
GASVM	A hybrid of **G**enetic **A**lgorithm and **S**upport **V**ector **M**achine
GP	**G**eometric **P**rogramming
GSA	**G**ravitational **S**earch **A**lgorithm
GSAPSO	A hybrid of **G**ravitational **S**earch **A**lgorithm and **P**article **S**warm **O**ptimization
GWO	**G**rey **W**olf **O**ptimization
HS	**H**armonic **S**earch
HWPSO	A **H**ybrid of **W**hale **O**ptimization **A**lgorithm and **P**article **S**warm **O**ptimization
IA	**I**mmune **A**lgorithm
IC	**I**ntegrated **C**ircuit
ICMR	**I**nput **C**ommon **M**ode **R**ange
IP	**I**ntellectual **P**roperty
LDO	**L**ow **D**rop-**O**ut
LG	**L**oop **G**ain
LiR	**Li**ne **R**egulation
LoR	**Lo**ad **R**egulation
MEMS	**M**icro**e**lectro**m**echanical **S**ystems
mGWO	**M**odified **G**rey **W**olf **O**ptimization
MPI	**M**essage **P**assing **I**nterface
NFL	**N**o-**F**ree-**L**unch
NMOS	**N**-channel **M**etal **O**xide **S**emiconductor
OPAMP	**OP**erational **AMP**lifier
OS	**O**ver-**S**hoot
PMOS	**P**-channel **M**etal **O**xide **S**emiconductor
PSO	**P**article **S**warm **O**ptimization
PSRR	**P**ower **S**upply **R**ejection **R**atio
PVT	**P**rocess **V**oltage **T**emperature
QR	**Q**ualitative **R**easoning
RF	**R**adio-**F**requency
SA	**S**imulated **A**nnealing
SCA	**S**ine-**C**osine **A**lgorithm
SF	**S**low **F**ast
SI	**S**warm **I**ntelligence
SoC	**S**ystem-**o**n-**C**hip

SPICE	**S**imulation **P**rogram with **I**ntegrated **C**ircuit **E**mphasis
SR	**S**lew-**R**ate
SS	**S**low **S**low
SVM	**S**upport **V**ector **M**achine
TS	**T**abu **S**earch
TSMC	**T**aiwan **S**emiconductor **M**anufacturing **C**ompany
TT	**T**ypical **T**ypical
UGB	**U**nity **G**ain **B**andwidth
UMC	**U**nited **M**icroelectronics **C**orporation
US	**U**nder-**S**hoot
VHDL	**V**ery High Speed Integrated Circuit **H**ardware **D**escription **L**anguage
VLSI	**V**ery **L**arge **S**cale **I**ntegration
WOA	**W**hale **O**ptimization **A**lgorithm
WOA-mGWO	A hybrid of **W**hale **O**ptimization **A**lgorithm and **m**odified **G**rey **W**olf **O**ptimization
WSTS	**W**orld **S**emiconductor **T**rade **S**tatistics

List of Symbols

ub	Upper bound
lb	Lower bound
α	Alpha wolf
β	Beta wolf
δ	Delta wolf
ω	Omega wolf
j	Search agent number
X_p	Position vector of the prey
X	Position vector of the search agent
X_a	Position vector of the alpha wolf
X_b	Position vector of the beta wolf
X_d	Position vector of the delta wolf
X_r	Position vector of the random search agent
t	Present iteration
$t+1$	Next iteration
$X(t+1)$	Updated position vector of the search agent
D	Distance vector
D_a	Distance vector between search agent and alpha wolf
D_b	Distance vector between search agent and beta wolf
D_d	Distance vector between search agent and delta wolf
V_{DD}	Supply voltage
W	Width of the transistor
L	Length of the transistor
S	Aspect ratio of the transistor
V_{tn}	Threshold voltage of n-channel MOS Transistor
V_{tp}	Threshold voltage of p-channel MOS Transistor
V_{μ_p}	Charge-carrier effective mobility of p-channel MOS Transistor
V_{μ_n}	Charge-carrier effective mobility of n-channel MOS Transistor
C_{ox}	Oxide capacitance
λ	Channel length modulation
g_m	Transconductance
g_{ds}	Output conductance
C_L	Load Capacitance
R_{out}	Output Resistance

I_D	Drain current
I_b	Bias current
Hz	Hertz
V_{dsat}	Drain saturation voltage
A_v	DC gain
P_d	Power dissipation
f_{-3dB}	Cut-off frequency

Prefixes Used for Units

Giga	G
Mega	M
Kilo	K
Milli	m
Micro	μ
Nano	n
Pico	p
Femto	f

Chapter 1

Introduction

1.1 Overview

In the recent past, the advent of sub-micron technology has led to rise in the level of integration resulting in an entire System-on-Chip (SoC). The rapid growth in the shipments of semiconductors has been drastically increasing from 32.6 billion devices in 1978 to around 1075.1 billion devices in 2017 [1] . According to WSTS Inc [2], the growth of worldwide market was all time high with 468.8 billion dollars in 2018. The technology advancements in semiconductor industry and high level of integration demands the use of efficient Computer-Aided-Design (CAD) tools.

The analog circuitry occupies comparatively smaller portion in the IC, but its design is a complex and time consuming process when compared to the digital counterpart, resulting in increased overall time-to-market [3]. The design of analog circuits requires highly skilled designers as it is one of the major bottlenecks in IC design process. Automation of digital circuits has been successful over the past few decades due to its structured nature and high level of abstraction. Even though, the analog portion constitutes a smaller proportion of an entire mixed-signal SoC, its low level of abstraction makes the design automation a complicated task. The reduced pace in the evolution of analog design automation makes it more challenging field of research [4],[5].

In this work, the challenge of analog design automation is addressed as a way towards easing the design process. The proposed approaches are based on a set of tools for automating

parts of the design flow to improve the efficiency and in-turn reduce the complexity and overall time-to-market.

1.2 Analog Integrated Circuits

With increase in the advancements, there is an ongoing trend for the replacement of analog circuit with digital wherever possible to exploit the fruits of scaled down technologies. But, in some tasks such as converting the pure analog signals from real world to suit the digital world, the usage of analog circuits is inevitable. Some of the examples include sensors, microphones and analog frontend for transmitting or receiving Radio Frequency (RF) signals. Besides, some significant applications of analog circuits include bridging a gap between the real and digital realms using of Analog to Digital converters (ADCs) and Digital to Analog Converters (DACs) indispensable. These circuits become more critical in mixed signal IC's for high speed communications.

RF circuits, in modern wireless communications with frequency range in GHz, are considered to be one of the most significant applications of analog circuits. Another important class of analog circuits include reference circuits providing a stable biasing conditions for not only the aforementioned circuits, but also pure digital circuits like clock generators and clock drivers. In all the above-mentioned circuits, amplifiers form one of the critical building blocks. This makes the analog circuit integration a necessary segment in large range of applications.

In recent years, besides being one of the revolutionary technologies in semiconductor and electronic sector, SoC technology has shown a remarkable progress. According to the report from Zion Market Research [6], the global mixed signal SoC market is expected to rise from 211.5 billion dollars in 2016 to 765.1 billion dollars in 2022, growing at a compound annual growth rate above 23.9 %. Thus, the analog circuits are, and will remain, the integral part of many ICs.

1.2.1 Analog Design Flow

The analog design flow starts with the implementation of an idea into a functionality, which is later mapped into an architectural realization. Firstly, the functionality is decomposed into a set of blocks until it can further be mapped into low level blocks. The functionality is validated by carrying out the simulations on high level models that set the target specifications on

the low-level analog building blocks. Performance specifications include various performance metrics of the circuit that act as the measures to characterize the behaviour of an entire cell. One of the examples is that an amplifier should have a reasonable unity gain frequency for proper operation.

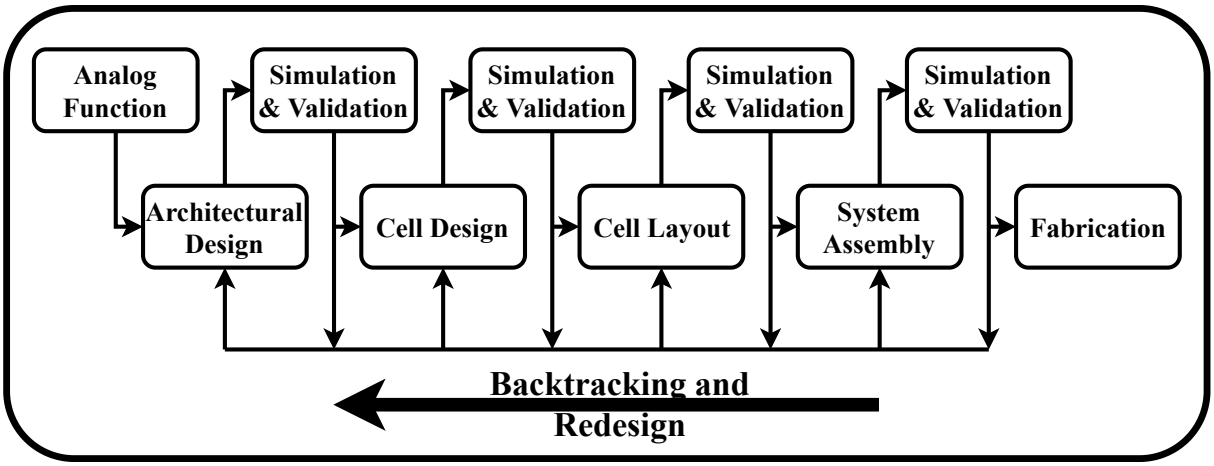


Figure 1.1: Analog Design Flow [7].

Further, the low-level building blocks are designed to comply with previously set performance specifications. The appropriate way of implementing the functionality is considered from different realizations in the cell design phase. Then, the layout is generated to set the geometries of the functional block. Final step is to assemble the building blocks to obtain the desired functionality. The entire design process undergoes simulation and validation at different stages. If the circuits fail to meet the desired specifications, the preceeding steps need to be revisited besides probable requirement of backtracking several steps for several times in the entire design process. The simplified analog design flow is illustrated in Figure 1.1. In this work, the major emphasis is on the design automation of analog circuits.

1.2.2 Manual Design Methodology

The manual design methodology of the low level design starts by applying the higher and/or lower limits to performance metrics. The traditional design flow, demonstrate in Figure 1.2, consists of three main phases, i.e., topology selection, circuit sizing and layout generation.

Topology Selection

Based on the design requirements, an appropriate circuit topology is chosen by the designer to meet the desired specifications. The designer has to select the best topology that satisfies the specifications from a set of topologies representing the same functionality. Even though techniques such as Qualitative Reasoning (QR) assists in reducing the number of possible candidates, there would still be several promising options left. The final topology is selected depending on the experiences, simple rules of thumb and hand calculations.

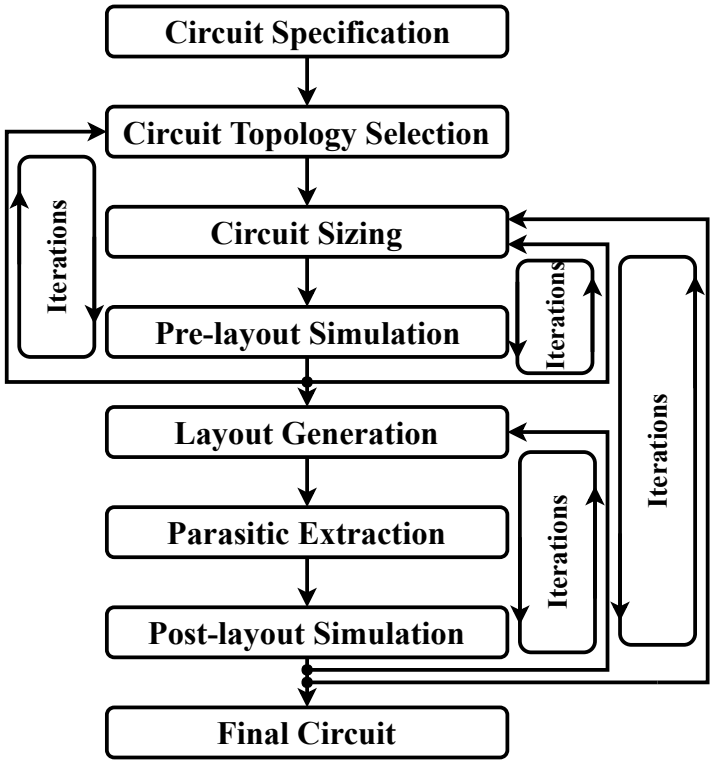


Figure 1.2: Flow for manual design of analog circuit.

However, the designer may also get influenced by several other factors apart from performance specifications, such as time limit, complexity of the circuit, familiarity of the topology and so on, resulting in unfavourable outcome of the design process.

Circuit Sizing

In this phase of design process, the bias voltages and currents are determined besides obtaining the sizes of circuit components such as transistors, resistors and capacitors. The nonlinear relation between various design variables and the resulting performance makes the entire

process a complicated task. The approximate sizes of the components can be calculated using simplified hand calculation models. These models are based on the transistor characteristics that may differ from real devices. In addition, previous experience of the designer and rules of thumb help in reducing the large design space.

The performance of the circuit is evaluated using a circuit simulator, such as, HSPICE^{TM} [8] or SPECTRE^{TM} [9], where a suitable set of inputs are applied to a testbench for obtaining performance metrics of interest. Initial set of design parameters do not ensure the satisfying performance specifications and hence require several adjustments. These tools aid the designers in adjusting the design parameters with their ability to sweep a value and determine its impact on the overall performance. However, sweeping all the design parameters increases the computational time. Thus, the search, in the number of parameters to be swept, is limited.

A simple form of performance optimization is applied in spectre by adjusting a few design parameters to fine tune the performance metrics. However, these routines can handle limited design parameters besides the requirement of a decent starting point resulting in minor improvements. Also, this operation requires more computation time.

Layout Generation

Layout generation is also considered as a significant part of analog design. Here, the devices after circuit sizing are projected onto a physical implementation for fabrication. The goal is to generate a compact layout considering parasitic effects and device matching. Extraction tools serve the purpose of evaluating the performance-shift by incorporating interconnect parasitics. This effect of parasitics should be considered by the designers to meet the performance specifications by adding some extra design margins. Resizing of the layout is a challenging task as the layout with smaller margin may include more iterations increasing the turn around time. Whereas, larger margins lead to larger chip area and increased power consumption.

1.3 Challenges in Analog Circuit Design

The design flow of analog circuits suffer from some major issues that will be discussed in this section.

Design of low-level cells is one of the most time-consuming tasks. As all the chips contain both analog and digital components, this acts as a severe bottleneck due to the gap between analog and digital design efficiency. Thus, the reduction in the design time of analog circuits in an IC has a major impact on the overall time-to-market of an entire chip. At the same time, the circuit performance cannot be compromised. While sizing a circuit manually, the designer has to consider tens of performance metrics and understand the effect of design parameters on the same. Despite of knowing all the relations, it is still difficult to track the change due to slight adjustments. There is a tradeoff among various performance metrics. Thus, comprehension of the problem becomes more complex for the designer with increase in the number of performance metrics.

The performance specifications are dependent on the design parameters of the circuit. When the desired specifications are reached, the results are finalized without exploring better options from the search space in manual design process resulting in the insufficient utilization of process technology. Consider two designs of an amplifier that satisfy all the desired performance metrics but with different power consumptions. The amplifier with minimum power consumption is preferred sensing its advantage. However, since every design in traditional design flow is costly, the number of circuits that can be examined for minimum power consumption, is limited. Suppose there is a restricted parameter space, shown in Figure 1.3, to

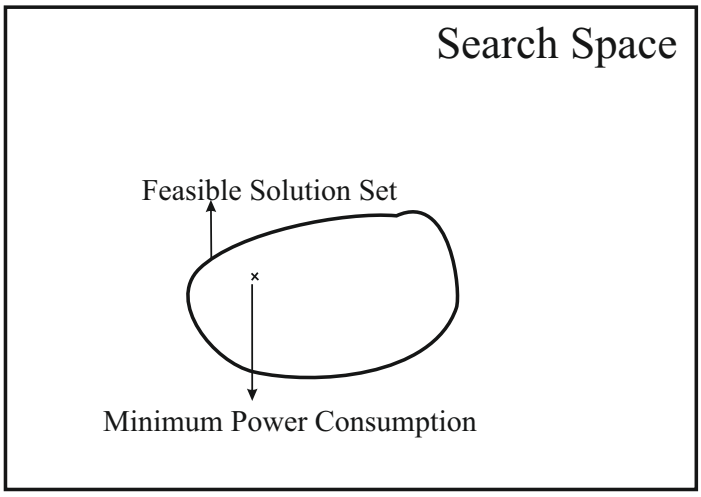


Figure 1.3: Design parameter space.

choose the device sizes from. With increased iterations, the design parameters are adjusted to meet the specifications by visiting different points in the entire design space. Continuation of the process would result in a part of design space where performance specifications are met. Further exploration of the feasible design space leads to the optimum power consumption.

Owing to the nonlinear relation of design parameters and performance metrics, the probability of obtaining the global or even local optimum solution is low. Furthermore, the exploration is limited only to a small fraction of the design space using manual design due to time constraints. To reach the full potential of the process technology, various circuit topologies have to be investigated. However, this is often overlooked due to the cost of designing several circuit topologies.

The design parameters change while porting the design of same circuit from one technology to another, mostly due to the constraints imposed by their process parameters. This demands the entire redesign of the circuit, making the level of analog circuit reuse comparatively lower.

To conclude, the limitations of the manual design are as follows:

- Insufficient exploration of design space.
- High risk of errors being introduced.
- Difficulty in reusing previous designs.
- Time consuming and thus costly.

1.4 Computer-Aided-Design of Analog Circuits

The major objective of design automation is to support the designers by addressing the drawbacks of manual design methodology. The manual design contains several possible iterations that include time consuming adjustments and validations. Automation of digital circuits has been successful over the past few decades due to its structured nature and high level of abstractions. On the other hand, evolution of reliable tools for analog circuit automation is not mature enough.

The automation of the design flow reduces the design time from weeks to days or even hours [10, 11]. The major focus of this thesis revolves around the circuit sizing, considering its significance in the overall design process.

The major approaches for solving the sizing problem include knowledge-based approach and optimization-based approach.

1.4.1 Knowledge-based approach

The knowledge-based approach is used in the first automated device sizing schemes in mid-1980's. Here, a predefined design plan, consisting of design equations and design strategy for sizing of circuit components to meet the performance specifications, is used as shown in Figure 1.4.

Usually, the design equations are formulated to evaluate the performance specifications for

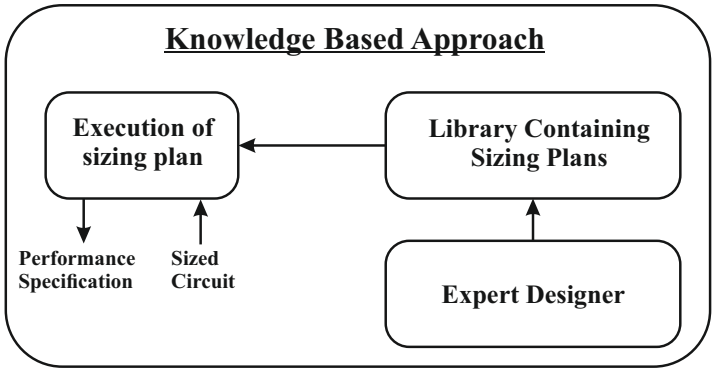


Figure 1.4: Knowledge-based circuit sizing approach.

given component sizes. In knowledge-based approach, the equations are restricted to compute device sizes from given set of performance requirements in order to keep the equation complexity at low level using simple device models.

In some cases, the number of design parameters exceed the number of performance metrics which makes the determination of parameters difficult by using equations. Thus, some of the design parameters are chosen using different ways to handle the extra degree of freedom. Knowledge-based approach uses a circuit specific design plan containing the information on choosing the design parameters to reduce the degree of freedom while achieving good performance.

1.4.2 Optimization-based approach

Optimization-based device sizing was developed to avoid the need for a topology specific design plan by increasing the generality of sizing tools. In this approach, the design plan is replaced by optimization algorithm. An optimization-based approach constitutes a circuit performance evaluator and an optimization engine.

Even though the major aim of optimization-based design tools is the same, they are classified into three different classes. One of them uses a set of equations that relate performance specification with design parameters known as equation-based approach. Another approach uses different neural network models to evaluate the performance of a circuit known as neural network based approach. The third approach uses SPICE like circuit simulator at every iteration to evaluate the circuit performance for a set of design parameters known as simulation based approach.

Equation-based approach

In equation-based approach, shown in Figure 1.5, equations derived manually or using symbolic analyzers are used to evaluate the circuit performance [12]. Here, the symbolic equations are derived from the netlist description. The accuracy of approach depends on the quality of the equations used. Manually derived equations are too simple compromising the accuracy of the tool when compared to models in the circuit simulator. This approach offers short setup and execution times at the expense of accuracy.

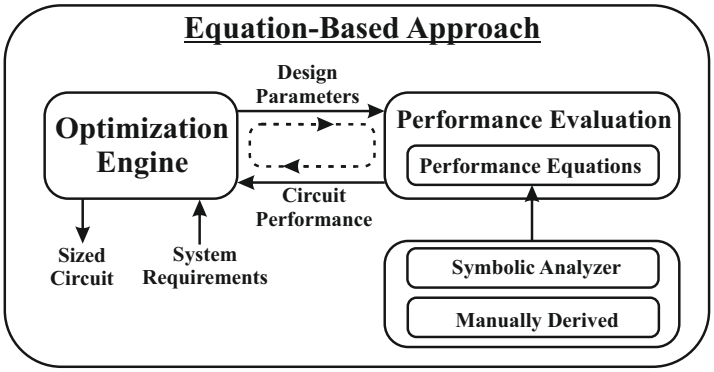


Figure 1.5: Equation-based circuit sizing approach.

Neural network based approach

Neural network-based approach models the behaviour of the circuit by training a neural network resulting in faster evaluation of performance metrics. However, a sufficient number of training samples in the region of interest are required. The neural network-based approach is demonstrated in Figure 1.6. There exists a trade-off between the accuracy of the neural network-based performance prediction and the amount of training data. However, achieving higher accuracy requires increased training data while increasing the time consumed by a high accuracy performance evaluator like a circuit simulator. This approach has short execution time

and large generality but suffers from accuracy problem and large amount of time on preparatory phase.

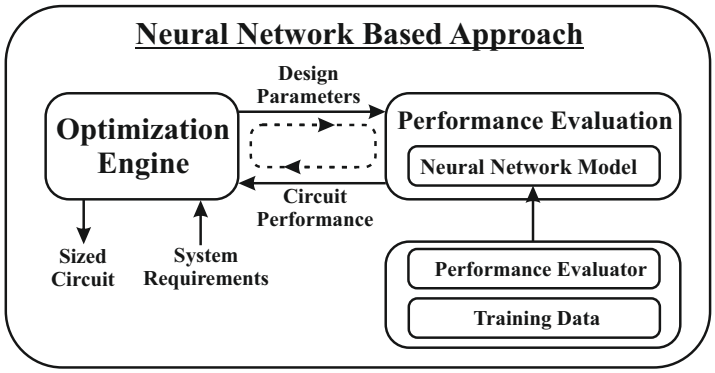


Figure 1.6: Neural network-based circuit sizing approach.

Simulation-based approach

Simulation-based sizing tool uses a circuit simulator and an optimization algorithm in a loop as shown in Figure 1.7. Their ability to include all performance metrics in the form of cost function aids in handling a varied range of analog circuits. This approach results in the accuracy equal to that of models used by the circuit simulator. The usage of circuit simulator

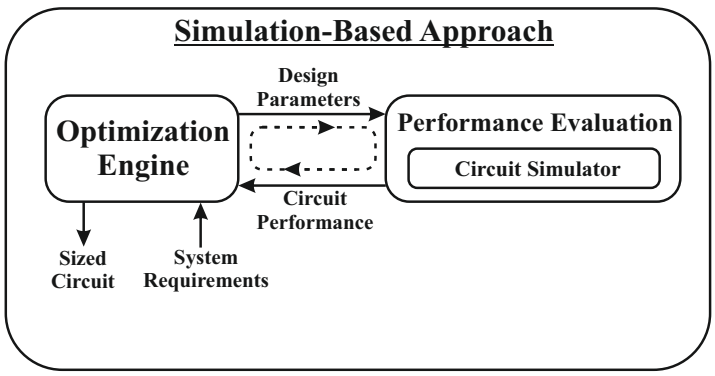


Figure 1.7: Simulation-based circuit sizing approach. [13]

in simulation-based approach helps in achieving higher accuracy. The approach results in high generality besides addressing a large range of design problems. However, it suffers from longer execution times.

1.4.3 Optimization Methods

The optimization phase in the design automation process consists of an optimization algorithm as a significant segment that considers the solution obtained in synthesis section to output an optimum solution. Optimization is a study of dealing with various problems in order to maximize or minimize one or more objective functions with some real or integer variables. The process of optimization is a systematic way of choosing appropriate value from the specified range. Of the defined limit, the major aim of optimization is to obtain the best possible value for the objective function. An optimization problem [14] can be better expressed as follows,

Given a function $f : S \rightarrow R$ from a set S to the set of real numbers with the goal of determining a solution $\overline{x_0}$ in S such that

$$\begin{aligned} f(\overline{x_0}) \leq f(\overline{x}) \quad \forall \ x \in S \quad for \ \ minimization \\ f(\overline{x_0}) \geq f(\overline{x}) \quad \forall \ x \in S \quad for \ \ maximization \end{aligned} \tag{1.1}$$

Here, S denotes the subset of Euclidean Space(R^n) which is a collection of equalities, inequalities or constraints that needs to be satisfied by the component of S. Where S is the search space and the member of search space is known as a candidate or a feasible solution. The search space is the domain of objective function or cost function (f). A feasible solution set contains one or more optimal solutions that optimizes the given objective function.

The optimization techniques applied in analog circuit sizing tools can be classified into two types, i.e., deterministic and metaheuristic optimization techniques. Deterministic optimization algorithms encompass traditional methods like Levenberg-Marquardt and Newton techniques that suffers from three significant issues, i.e., getting trapped into local minima, dependence on differentiability and continuity of the objective function, and it requires a decent starting point [15].

In contrast to the deterministic approach, metaheuristic algorithms have better capability and efficiency in dealing with complex optimization problems. Metaheuristics are classified into stochastic search, evolutionary and swarm intelligence algorithms. The stochastic search algorithm is a discrete or continuous local and global search process, which does not require continuous values for convex or differentiable cost functions. Applications of the stochastic search process, in designing of analog circuits, such as Simulated Annealing (SA) and Tabu Search (TS) include OPTIMAN [16], GBOPCAD [17], Severo [18] and Bland [19], respectively. Evolutionary algorithms are general population-based metaheuristic algorithms which

utilize biologically inspired mechanisms such as reproduction, mutation, recombination, and selection. The algorithms employed for the design of electronic circuits that come under the category of evolutionary algorithms are Genetic Algorithm (GA) [20, 21, 22, 23, 24, 25], Genetic Programming (GP) [26, 27, 28, 29], Differential Evolution (DE) [30, 31] and Immune Algorithm (IA) [32, 33]. Swarm Intelligence (SI) is a cooperative intelligence behavior of decentralized and self-organized systems with fundamental yet essential concepts of self-organization and division of labour. Swarm intelligent algorithms that are being used for circuit sizing and optimization include Ant Colony Optimization Algorithm (ACO) [34, 35, 36], Particle Swarm Optimization (PSO) Algorithm [37, 38, 39, 40, 41, 42], Gravitational Search Algorithm (GSA) [43, 44], etc. In recent past, swarm intelligence algorithms have gained significant interest, due to their robust nature, ease of implementation and high flexibility, especially for the development of analog circuit sizing tools.

1.5 Motivation

In the previous section, some of the issues related to manual analog design flow were discussed. These issues tend to grow more complicated while transition to the era of SoC. Owing to high level of abstraction and automation, digital demand is satisfied with short time to market. The scenario is different for analog circuits due to their low level of automation.

The analog designers rely mostly on circuit simulators and layout editors for design of analog circuits and generation of layout, respectively. These tools include very little automation. Hence, the analog design process is mostly based on human efforts. Whereas, the high level of abstraction in digital circuit design makes the digital design flow more suitable for automation. For example, programming languages such as VHDL and Boolean expressions can be implemented using only few logical building blocks. Therefore, standard cell libraries with limited blocks are sufficient enough to cover the majority of the digital circuits.

Higher number of performance specifications in analog design render the usage of standard cell libraries infeasible. However, creating libraries using analog circuits require large number of cells to cover a varied range of specifications in analog design. One of the major shortcomings of analog design automation is the lack of structured design flow. The designer has to consider different effects while choosing the appropriate topology. Nevertheless, the decision taken at one level effects the design in following levels. This complicated design flow

has led to the little success in qualifying the design parameters and performance specifications adding up to the complications in analog design process.

Another important factor that has to be considered is the time-to-market which is much higher for analog circuits when compared to its digital counterpart. Conversely, the market for analog circuits is much smaller than for digital circuits. This is responsible for the reduced interest in the research related to analog design automation. However, the advancements in the mixed signal SoC's have triggered the need for various analog design automation tools.

1.6 Objectives

Analog design automation may act as a solution to some of the problems in analog circuit design or can at least help in reduction of the others. The objectives of thesis are:

- To improve the conventional algorithms or develop new hybrid optimizing algorithms to enhance their exploration ability for faster convergence while avoiding local minima stagnation.
- Investigation of different techniques for design optimization of CMOS analog circuits using developed metaheuristic optimization algorithms.

1.7 Scope of the thesis

The main aim of the thesis is to develop a design automation methodology for different analog cicuits. The generic model assists in designing other types of circuits such as mixed signal and RF circuits. The long-term goal is to automate major part of analog circuit design incorporating longer design structures. Even though the entire design problem is worth addressing, this thesis addresses the device sizing problem through automation. The developed tool is able to derive a set of nominal design parameters by satisfying the constraints while meeting the required performance specifications. However, this tool is still in its infancy and needs further developments.

An original contribution of the work in the field of analog design automation is presented in the ensuing chapters of the thesis. Some of the major contributions include,

- Application of conventional optimization algorithms i.e., Whale Optimization Algorithm (WOA) and Grey Wolf Optimization (GWO) algorithms for transistor sizing problem of analog circuit design. Two amplifier circuits i.e., CMOS differential amplifier (DIFFAMP) and two-stage CMOS operational amplifier (OPAMP) is considered as test cases.
- The conventional GWO algorithm is enhanced to improve the exploration ability by incorporating the randomness in the optimization process. The same is applied for circuit sizing of the CMOS DIFFAMP and two-stage CMOS OPAMP.
- A hybrid of mGWO and WOA algorithm is proposed to increase the performance of the conventional algorithms. The WOA and mGWO algorithms are hybridized using low level teamwork hybridization methodology where in the two algorithms strive combinedly to obtain the global optimum solution. This hybrid WOA-mGWO algorithm is applied for the design of two-stage CMOS OPAMP. These methods use equation-based optimization methodology for circuit sizing.
- However, equation-based circuit sizing suffers from lower accuracy. Thus, a design methodology is proposed using the hybrid SCAWOA algorithm using simulation-based approach. This approach is used for design automation of folded cascode operational transconductance amplifier (FCOTA) for different technologies and adaptive biased, capacitor-less CMOS low dropout voltage regulator.

1.8 Organization of the work

The thesis presents analog device sizing methodology using new hybrid evolutionary algorithms. The thesis is distributed into seven chapters. The summary of the chapters is as follows:

Chapter 1 presents the detailed introduction followed by motivation, problem statement and contribution to the thesis.

Chapter 2 gives the overview of various analog design automation and optimization methodologies.

Chapter 3 describes the application of conventional optimization algorithms such as GWO and WOA for the design of CMOS amplifier circuits.

Chapter 4 demonstrates the proposed enhanced grey wolf optimization algorithm and its application to circuit sizing problem of CMOS analog circuits.

Chapter 5 discusses the new hybrid WOA-mGWO algorithms and its application to circuit sizing problem of two-stage CMOS OPAMP.

Chapter 6: Presents a novel circuit sizing strategy using simulation-based optimization with SCA-WOA algorithm as an optimization engine which is applied for the design automation of FCOTA and capacitor less LDO.

Chapter 7: Provides the summary of the work and future work to further sophisticate analog design automation tools.

Chapter 2

Literature Survey

2.1 Introduction

As discussed in chapter-1, there is a need for design automation to address the limitations of manual design and simultaneously aid analog designers in designing complex and time consuming analog circuits. In this chapter, a detailed review of existing tools describing the basic principles of alternatives to manual design methodology is presented. As the major aim of this thesis is to automate the circuit sizing process of analog IC, various methods to accomplish the same are discussed in detail.

2.2 Review of CAD tools and methodologies

Traditionally, there are two main approaches for the circuit sizing of analog circuits, viz., knowledge-based and optimization-based. In the knowledge-based approach, the predefined design plan is used to obtain the design parameters of the circuits. On the other hand, the optimization-based approach uses different optimization mechanisms to complete the task. The optimization-based approach is categorized into equation-based, neural-network-based and simulation-based methods.

2.2.1 Knowledge-based approach

IDAC [45], one of the earliest and popular knowledge-based design tools, was developed in the Centre Suisse d' Electronique et de Microtechnique, Switzerland. As it is a knowledge-based tool, it relies upon a set of circuit-specific design plans for different topologies. IDAC supports a varied range of circuits including voltage and current references, amplifiers, comparators and ADCs. Expert designers are required to create a library containing an organized set of design equations. If the circuit fails to satisfy the design requirements, after verifying the performance using the design plan, the tool adjusts the specifications and executes the design plan again. This procedural execution results in shorter design times for circuits already present in the database. However, besides a large library of design plans, it is difficult to incorporate different design plans for a varied range of topologies present in analog design while requiring the designers to slightly modify the topology to achieve the desired performance. This necessitates the development of a new design plan for a wide range of possible scenarios. Moreover, the overall performance of the tool depends on the quality of the design plan and design equations used. This method is limited to simple models resulting in compromised accuracy.

Another tool with knowledge-based design mechanism is OASYS [46], [47] that was developed at Carnegie Mellon University in Pittsburgh, USA. Here, the circuit is partitioned into different sub-blocks. The majority of the circuits using this tool are amplifiers that are decomposed into sub-blocks such as voltage references, current mirrors and so on. The process of sizing starts by choosing the topology based on performance specifications and then dividing it into sub-blocks and deriving corresponding performance specifications. The tool picks the sub-block with the best performance from a set that offers the same functionality. At the bottom level, the tool uses simple device models to determine design parameters. The tool uses the possible backtracking strategy to deal with the possible discrepancy in the estimation of low-level block performance. Besides the usage of simple models, it also requires a long time to create a design plan. According to [47], the time used for the creation of a design plan for a two-stage operational amplifier was 18 months. The design of fundamental sub-blocks increases the level of its reuse. However, it is still a challenging task to generalize the sub-blocks.

BLADES [48] is one of the earliest tools to use artificial intelligence (AI) for partitioning and sizing of analog circuits to mimic the behaviour of an expert analog designer. A divide and conquer methodology is applied where several sub-circuits, such as a differential amplifier, gain stage and output stage, are combined to form an operational amplifier. It contains approximately 250 different rules applied combinedly for operations like partitioning and sizing. Besides the

combination of rules, it also uses look-up tables obtained using simulation results. However, similar sizing of transistor-level blocks results in reduced accuracy besides requiring the storage of look-up tables for different device models, manufacturing process and specifications.

Another tool based on Qualitative Reasoning (QR) to adjust the performance of the circuit is ISAID [49] that is developed at the Imperial College in London. Qualitative reasoning is the procedure to describe the changes or mechanisms in the physical world. This method replaces the exact performance relations with quantitative relations, i.e., describing the effect of variation of design parameters on the performance of the circuit. While varying the design parameters, the sign of the gradient of performance metrics is considered to understand the effect of changing a specific parameter.

The knowledge-based approach is a time efficient procedure of designing any analog circuit with an overhead of lower accuracy due to usage of simple design equations and creation of design plan makes it an inefficient procedure of design for modern technologies. However, development of separate design plan for individual topology makes this approach less reusable and thus reduces the generality.

2.2.2 Optimization-Based Circuit Sizing

The Neural Network-Based Approach

One of the popular approaches for circuit sizing using neural network models is devised by Alpaydin [50, 51]. Here, the performance metrics are obtained using a neuro-fuzzy model which is trained using the test samples from SPICE simulator. The neuro-fuzzy model approximates the behaviour of linear or non-linear circuits. The accuracy of the approach depends on the number of test samples covering the search space. Thus, to achieve higher accuracy, a large set of training samples are required. Besides, it also uses manually derived equations for evaluating other AC performance metrics. The approach uses SA as an optimization methodology.

An Artificial Neural Network (ANN) based CAD framework is presented by Kaustubha Mendhurwar et. al. [52] for sizing of MOS transistors in analog amplifier circuits. The process starts with the selection of building blocks based on specifications followed by evaluation of design parameters using ANN binning and correction models. Finally, constraints are applied before presenting different design choices as output. It uses HSPICE for creation of a set of test samples. Another ANN based approach for circuit sizing is reported in [53] that starts with

basic building blocks like current mirrors and differential amplifiers. The same methodology is applied to various digital circuits in [54]. Other variants use macro modelling [54] and surface modelling [55] as optimization methods for sizing of analog circuits. Macro modelling is a circuit-specific development of models using the mathematical equivalent of various internal functions, considering the overall circuit as a black box. Whereas, surface modelling is a process of selective evaluation of response surface model which is also a circuit level design methodology that is repeated for every new circuit. Similar approaches have been presented in [56] that also involve simulation and/or optimization for the circuit sizing of every new circuit.

Neural network-based approach suffers from lower accuracy owing to the usage of finite data samples for evaluation of design parameters. Also, every topology requires a separate set of manually derived equations besides requiring separate set of test samples for individual circuit or topology even if there is a slight modification in the circuit. However, most of the optimization phases use local search algorithms that have the risk of getting trapped in local minima.

Equation-Based Approach

OPASYN [57] is one of the equation-based design methodologies that includes topology selection, circuit sizing and layout generation completing the basic design flow. It consists of a database containing each step for certain selected circuits. Analytic circuit models are used for performance evaluation in the optimization phase. The models are specifically developed for amplifier circuits with database consisting of bounds for design parameters and manually derived equations besides independent design parameters. The models have 200% of error over HSPICE simulations which is reduced by addition of fitting parameters. However, the error still remains above 20%. The steepest descent algorithm is used for optimization by selecting appropriate starting point to improve the performance of the algorithm.

STAIC [58] is a tool developed at the University of Waterloo in Canada that provides a framework to find design trade-offs while exploration within a reasonable time. It divides component sizing into two parts. In the first step, a grid-based scan is performed over the entire design space using simple device models to visualize the trade-offs. Next, the result of the scan is used as a starting point for secondary optimization using more accurate models from simulation-based approach. Thus, STAIC is mainly used to provide an initial point.

Maulik et.al. [59, 60] have proposed a tool for performing topology selection and circuit sizing, simultaneously. Here, a branch-and-bound optimization technique is used to find the suitable topology for determining component sizes. In contrast with previous approaches, this tool uses a relaxed DC formulation by considering it as a part of cost function. Thus, the circuits may not be physically feasible that are visited during the optimization run. In this tool, high level BSIM [61, 62] models are used for achieving high accuracy while obtaining design parameters. However, the small signal performance of the circuit is evaluated using manually derived equations.

ASTRX/ OBLX [63] is a tool that relies on encapsulated device and asymptotic waveform evaluators (AWEs) while performing a relaxed DC formulation besides using SA as an optimization engine for device sizing optimization. This tool uses both simulation-based and equation-based approaches. To improve the accuracy and to simultaneously reduce the simulation times, ASTRX/OBLX uses AWE which is an efficient approach for analysing linear circuits. In AWE, a reduced complexity model is used to predict the small-signal performance metrics and other performance metrics are computed using circuit equations which makes it considerably faster than circuit simulators like SPICE. However, the AWE approach is not suitable for modelling non-linear circuit behaviour. Furthermore, the accuracy is compromised due to the usage of lower order model as circuit transfer function. Similar evaluation is also used in DARWIN which uses genetic algorithm in the optimization engine [20].

AMGIE [64, 65] is a tool that deals with all phases of circuit design including topology selection, circuit sizing and layout generation. The circuit sizing is performed using several tools to create a design framework. A symbolic analyser is used to derive small-signal equations from AC analyses of circuit topology, which are further reduced to be evaluated in the optimization loop. Eventhough the system supports the automated setup of design equations, it is mostly intended for the expert users where they create libraries containing topology specific sizing plans. The optimization supports both global and local optimization methods such as SA and gradient-based methods, respectively.

GPCAD [66, 67] is a tool developed specifically for device sizing of operational amplifiers using geometrical programming (GP). The design equations are expressed in the form of posynomial equations resulting in convex optimization problem that is formulated using cost function and inequality constraints while equality constraints are expresses in the form of monomials. However, using posynomial based design equations results in reduced accuracy as high accuracy device models cannot be replaced by the posynomials. Furthermore, the tool does not

automate the process of obtaining these equations limiting the usage of the tool. Thus, it offers reduced execution time at an expense of accuracy [13].

SD-OPT [68] is specific tool for designing switched-capacitor delta-sigma modulators using two-stages of optimization. Firstly, a behavioural simulator explores the available design space by operating on the library of design equations while obtaining initial sizes at the modulator level. The output of the first stage sets the performance specifications for the cell-level design. The second stage is a simulation-based approach is used for cell level sizing of the modulator. The optimization method in both these stages is SA. This tool demonstrates an efficient way of designing modulator circuits. However, addition of new topologies for modulators and SC circuits still remains a time consuming and cumbersome task.

Alex Doboli et. al. [69] presents a methodology for synthesis of analog circuits resulting the netlist of analog components and sized simultaneously to optimize the specified objectives. The process of automation consists of two steps, wherein, first step deals with architecture generation using branch-by-bound algorithm and followed by component sizing and constraint transformation using genetic algorithm.

Kazuo Matsukawa et. al. [70] developed a tool to design two ADC architectures, i.e., Pipeline ADC and continuous time Sigma Delta ADC via convex optimization by relating the performance of the converters with component sizes. Convex optimization is a process of solving the expressions representing circuit performance with many design variables.

A tool [71] was presented by Lui S. et. al. for designing analog circuits over nonconvex polynomial objective function and constraints, which is then solved using convex programming techniques. This framework considers both equality and inequality constraints for accurate device modelling and parameter tuning. This method is used for the design of sigma delta ADC and nested transconductance – capacitance compensation amplifier circuits.

In [72], M Fakhfakh et. al. presents an equation-based design methodology for the design of analog circuits using Particle Swarm Optimization (PSO) algorithm. It is used for the maximizing the voltage gain for low noise amplifier and maximize the high current cut-off frequency while minimizing the parasitic input resistance of a second-generation current conveyor. Kotti et. al. [73] demonstrates the comparison of two algorithms, i.e., PSO and ACO algorithms, for device sizing of analog circuits. The benchmark circuits are second generation current conveyor (with minimum parasitic input resistance and maximum high cut-off frequency as objectives)

and two-stage CMOS operational amplifier (with voltage gain, common mode rejection ratio, transistor area and power dissipation as objectives).

R A Varul et. al [74] investigates the performance of different evolutionary algorithms, i.e., Genetic algorithm, ABC and PSO algorithms, on active analog filter design for selection of passive components with respect to accuracy and execution time. It evaluates the algorithms by considering their own internal parameters to obtain minimum design error. The circuits used for validation are fourth order Butterworth low-pass analog filter and second order state variable active filter design. The same circuits are used to validate other metaheuristic algorithms i.e., DE and Harmonic Search (HS) [75]. A design optimization technique [76] was also developed for circuit sizing of analog circuits, i.e., CMOS DIFFAMP with current mirror load and two-stage CMOS operational amplifier, using PSO algorithm. The objective considered here is to optimize the over MOS transistor area of the benchmark circuits while satisfying the performance constraints. They also investigate the usage of PSO algorithm for electronic circuit design. The performance of the algorithm is evaluated for the design of an inverter while considering three different cases with different ranges and constraints of design criteria over transient response. The objective considered is to obtain minimum error for all the test cases [77].

S. Dam et.al. [78] discuss the equation-based design methodology for sizing of analog circuits using hierarchical abstraction, which is a process of translating the specifications from high level blocks to lower levels. Here, the hierarchical abstraction is associated with Geometric programming-based CMOS circuit sizing method to design a 4th order Sallen-Key low-pass filter using top down methodology. The major objective is to reduce the design time while improving the overall accuracy of the design.

In [79], Mallick et. al. proposes a combination of two population-based metaheuristic algorithms, i.e., GSA and PSO, to overcome the issue of suboptimality in individual algorithms. The GSA-PSO algorithm is used as an optimization technique in the optimization engine of equation-based design methodology for optimal design of analog circuits with an objective of obtaining minimum overall MOS transistor area while improving the overall performance of the circuit. The circuits considered for design optimization are CMOS DIFFAMP with current mirror load and two-stage CMOS operational amplifier. Similar tool [80] is also proposed to optimize the design of the same analog circuits, i.e., CMOS DIFFAMP and two-stage CMOS OPAMP, to obtain minimum overall MOS transistor area while satisfying the performance constraints. A hybrid of two algorithms, i.e., backtracking search and DE algorithm, is used as an optimization algorithm in the optimization section of the equation-based design tool.

Bishnu Prasad De et. al. [81] presents an approach to design and optimize analog circuits using a hybrid of PSO and ALC. The analog circuits that are used to validate the approach are CMOS two-stage comparator considering PMOS input driver with n-channel input and a CMOS folded cascode operational amplifier. The objective considered for the design of the abovesaid circuits is to obtain minimum over MOS transistor area. The same circuits are used to validate the approach using colliding bodies optimization (CBO) with reduction in the MOS transistor area as objective while satisfying the performance constraints. [82]

A modified version of conventional PSO algorithm known as craziness-based PSO was proposed by S Mallick et. al [83], which adopts various random variables for faster and better exploration and exploitation of the search space. Combination of craziness factor while deriving the velocity in PSO improves the exploration making it less vulnerable to local optima stagnation. This algorithm is used in the optimization phase of equation-based design methodology for circuit sizing of analog circuits, i.e., CMOS DIFFAMP with current mirror load and two-stage CMOS OPAMP, with an objective of obtaining least MOS transistor area.

Kanchan Baran Maji et. al [84] presents an equation-based design methodology using a swarm intelligence-based optimization algorithm, i.e., fish swarm optimization algorithm (FSOA). It is applied for the design optimization of analog circuits, i.e., CMOS two-stage comparator and CMOS FCOTA, while satisfying the design constraints on various performance metrics. The objectives considered for the design of analog circuits are minimum MOS transistor area, high gain and low power consumption. The same analog circuits are also used to validate the design optimization approach using simplex PSO algorithm. The simplex PSO algorithm is a hybrid of Nelder-Mead Simplex method and PSO algorithms that does not consider the velocity of the particle expression from conventional PSO algorithm [85].

A. B. de Andrade et. al [86] uses a constrained optimization for the design of bandgap reference circuits considering mismatch errors to satisfy the given accuracy specifications while aiming at obtaining minimum MOS transistor area. The process starts with the development of the initial design to create boundary conditions followed by characterization of main mismatch errors and finally performing the optimization to obtain minimum area. A simple trim circuit is also incorporated to demonstrate the reduced inaccuracy with negligible increase in area.

C. L. Singh et. al [87] combines two algorithms, i.e., PSO and aging leader and challenger algorithm, for exploration of the search space and computation of constraints. This algorithm is used in the optimization section of equation-based design methodology for optimization of

analog circuit, i.e., low-noise CMOS DIFFAMP with current-mirror load with thermal noise and overall MOS transistor area as objectives.

In [88], Bishnu Prasad De et. al. explores the application of two variants of conventional PSO algorithm, i.e., craziness-based PSO and hybrid ALCPSO, for design optimization of nulling resistor compensation-based CMOS two-stage operational amplifier circuit. Both the algorithms are used for design optimization with minimum MOS transistor area as objective while satisfying the performance constraints. The results from both the techniques demonstrates that ALCPSO is much better than CRPSO in terms of MOS area, overall gain and power dissipation. They have also proposed an efficient equation-based design technique [89] for two analog circuits, i.e., CMOS DIFFAMP with current-mirror load and two-stage CMOS OPAMP circuits. The conventional DE algorithm is also hybridized with random PSO to improve its performance. The proposed hybrid reduces the uncertainty and sub-optimality issues from DE and random PSO algorithms, respectively, demonstrating its superiority over conventional algorithms. The objective considered for optimization is overall MOS transistor area while maintaining the performance and design metrics within the specified limits.

A methodology [90] was proposed by Paramita Sarkar et. al. for design optimization of two-stage CMOS operational amplifier for obtaining minimum area and offset voltage while meeting various design specifications such as Gain, Slew Rate, etc. Moreover, offset minimization technique is also applied for further reduction of the offset voltage while maintaining current imbalance at the output stage of the amplifier. The optimization algorithm used in the optimization phase is Whale Optimization Algorithm.

A new population-based hybrid of whale optimization algorithm and PSO for solving complex optimization problems was proposed by Naushad Manzoor Laskar et. al. [91] The HWPSO overcomes the limitations of conventional algorithms, i.e., the risk of stagnation at local optimum solution. It is based on two phenomena, i.e., forced WOA that guides the PSO in avoiding local minima and capping phenomenon that deals with limiting the search space of WOA in the exploitation phase, to obtain better solution close to global optima. The proposed algorithm is used for solving three electronic design problems, i.e., analog circuit sizing of two-stage CMOS OPAMP, pull-in voltage minimization in RF MEMS switch and random offset minimization in CMOS DIFFAMP.

Simulation-based optimization

Simulated-based circuit sizing has gained a significant interest in the field of analog design automation considering its high accuracy while the availability of high computing resources has paved a way towards reduction in the overall computation time for simulation-based circuit sizing. AIDA-C [92] is one of the simulation-based tools that uses a circuit simulator to evaluate the circuit performance besides an optimization engine. However, Early approaches have used local optimization algorithms, such as SA, in the optimization engine of simulation-based design methodology. These tools have an advantage of handling large variety of analog circuits by including number of performance metrics while evaluating the cost function. Also, the performance accuracy is equal to the models used in the simulator making it one of the preferred circuit sizing when accuracy is the concern.

One of the commonly known simulation-based design tools is DELIGHT.SPICE [93] that is developed by University of California, Berkeley, USA. It uses SPICE as a circuit simulator and DELIGHT as an optimization toolbox. The optimization algorithm used by this tool is a method of feasible directions, which is an aggressive optimization methodology which directs the search based on design constraints and the worst performance. However, this tool requires several hours to perform the operation as it executes an optimization loop containing circuit simulator and optimization engine. Besides, it requires a decent starting point, when a large search space is considered, to start the optimization process due to divergence problems in SPICE simulator. Thus, this tool is more suitable to fine tune the manual designs.

Kuo-Hsuan et. al. [94] present a framework for synthesis of analog circuits facilitating the tradeoff aspect identification and optimal specification setting. The hierarchical process starts with characterization of the device followed by mapping of circuit level parameters from geometry/biasing parameters to obtain performance metrics and finally fine tuning obtained design parameters through reverse identification. Here, the equation-based circuit sizing is used to derive at an approximate solution and simulation-based approach to optimize the design.

FASY [95, 96] is a tool that uses a fuzzy-logic based topology selection and a two-phase optimizer is used on the selected topology to obtain the optimal solution while satisfying the performance constraints. In the two phase optimization process, the first step deals with simple analytical models for circuits and devices with SA as optimization algorithm are used to obtain the solution of the cost function and then, considering this solution as a starting point, a standard conjugate gradient algorithm in addition to SPICE models are used to evaluate the

circuit performance while obtaining the final performance. This tool can design a varied range of CMOS operational amplifiers.

In [97], Cheng Wu et. al. proposed a design methodology that is implemented in C language and SA. However, a decent starting point is needed for the proper operation of the optimization engine using SA. This starting point is generated by gm/id design methodology. Here, the SPICE netlist is generated based on the lookup tables and instead of finding the sizes of the transistors, this technique uses biasing conditions to solve the circuit.

FRIDGE [98] is one of the popular simulation-based optimization tools that also uses SA as its optimization engine. However, the cooling scheme is modified to replace the monotonical and slow reduction of temperature, used in the conventional SA, with adaptive cooling scheme with a series of fast cooling and reheating while decreasing the iteration with about 6 times on average. The process of optimization is divided into two phases. In the first phase, the design parameters are quantized according to the grid and then each node is evaluated while saving the results obtained to avoid visiting of the same node several times. In the second phase, after finishing the global search process, a gradient-based optimization algorithm is used to fine tune the overall performance of the circuit. Similarly, [99] Castro-Lopez et. al. proposed a tool that uses SA for optimization and deterministic approach for fine tuning the obtained performance of the circuit.

Barros et. al. [100, 101, 102] presents a circuit sizing optimization mechanism that uses both circuit simulator and automatically trained support vector machine to evaluate the performance of the circuit. The optimization algorithm used in the optimization engine of the proposed tool is modified GA. This mechanism supports both equations and simulation nased evaluation engines to evaluate the cost functions using behavioural models, based on either SVM or using electrical simulation. Here, SPICE like simulator is used for evaluation of performance specifications. Moreover, a GA-SVM learning scheme is also used for circuit sizing of amplifiers. R. Santos-Tavares [103] presents a framework for time domain optimization of amplifier using parallel GA based on Message Passing Interface (MPI), i.e., using multiple processors, simultaneously, for performing a complex computation task to reduce the overall execution time.

The tools, ANACONDA [13] and MAELSTROM [104], share same optimization framework with the only change being the optimization algorithm used. ANACONDA uses stochastic pattern search as an optimization algorithm whereas MAELSTORM uses GA/SA. Both the tools were developed by Carnegie Mellon university, Pittsburgh, USA. They use a wrapper to in-

terface an optimization with the circuit simulator. Similar to previous tool, the optimization task is distributed over a cluster of workstations for reducing execution time of complex optimization problems. However, the resulting performance of the tool depends on the initial starting point provided to the optimization algorithms indicating the necessity of evaluating several starting points for sufficient exploration of the search space.

Ali Jafari et. al. [105] presents novel constraint-based hybrid shuffled frog leaping algorithm with new leaping principle, for computer aided design of analog ICs. Also, instead of random evaluation, mutation is incorporated to generate new search agents to improve the convergence velocity of the algorithm. The optimization engine and SPICE simulator are linked together for simultaneous evaluation of the circuit performance. The analog circuits considered as benchmark are nested miller compensated three stage CMOS operational amplifier and double pole-zero cancellation compensated operational amplifier.

In [106], Kuber M et. al propose a novel optimization tool implemented in cadence design environment resulting in short setup time for optimization. The optimization algorithm used for the optimization engine is robust differential evolution algorithm. The automation is enhanced by optimization watchdog feature with the ability to automatically change bounds of the search space. The proposed tool is used for the design automation of two-stage miller operational transconductance amplifier, FCOTA and voltage regulator.

Mansour Barari et. al. [107] investigate the design automation of analog circuits using two optimization algorithms, i.e., GA and PSO. This tool also links SPICE simulator and optimization engine in the optimization resulting in a simulation-based optimization methodology. Firstly, the GA algorithm executed to result in a primary input to the SPICE simulator and then the output of SPICE simulator is given to the PSO algorithm for optimization which combinedly forms the optimization loop. The analog circuits used to validate the proposed techniques are two-stage CMOS operational amplifier and folded cascode operational amplifier.

Maryam Dehbashian et. al [108] present a novel technique for automated sizing of analog ICs using Advanced hybrid of gravitational search algorithm and PSO algorithm as an optimization engine. Also, a technique named shrinking circles is used to balance the exploration and exploitation abilities of the optimization algorithm. This tool uses the SPICE simulator as the evaluation engine and is interfaced with the optimization engine in the optimization loop. The testcases considered for the validation of the proposed tool are two-stage CMOS operational amplifier and folded cascode operational amplifier. Maryam Dehbashian et. al. They have also used the same tool with change in the optimization algorithm used in the loop from

hybrid AGSAPSO to advanced GSA algorithm. It also uses shrinking circles techniques with AGSA for revisiting the balance between exploration and exploitation of the optimization algorithms. A two-stage operational amplifier is used as the benchmark for validating the proposed technique with power consumption as an objective [109].

2.2.3 Commercial device sizing tools

In chapter 1, various reasons for the reduced interest in the analog design automation when compared to its digital counterpart is discussed. However, there is a vivid change in the market scenario owing to the increase in technology scaling and hence the design complexity. With current EDA tool-sets which are based mostly on the manual methodology, analog CAD represents a huge market opportunity with the potential to become one of the hottest market segments in EDA. The tools discussed above are the result of the research in the field of analog design automation and are classified as university tools. However, there are few companies that offer device sizing which are mostly commercial versions of university tools. Some of these tools are discussed below.

ANACONDA and MAELSTROM have been enhanced and converted to commercial versions resulting in a tool named NEOLINEAR [110] which was founded by the research group behind ANACONDA and MAELSTROM. This tool offers a design suit for analog device sizing, RF device sizing and layout in the form of NeoCircuit^{TM}, NeoCircuit-RF^{TM} and Neo-Cell^{TM}, respectively. These tools are simulation-based and are interfaced with industrial standard simulators. Moreover, they help in successful design of high-performance analog ICs in industry [11, 12] resulting in reduced design time from weeks to days. Analog design automation [111] is similar to NeoCircuit^{TM} and is also based on simulation-based circuit sizing methodology by interfacing it with industrial standard simulators.

Barcelona Design [112] is founded by the design group behind GPCAD that offers synthesizable Intellectual Property (IP) blocks containing the required posynomial-based design equations. That is, a customer is licensed with a circuit specific optimization engine. The optimization engine uses Geometric Programming for approaching the global optimum solution. The synthesizable IPs supplied by this company are amplifiers, data converters and Phase Lock-Loops. The tool is limited to three single ended topologies of operational amplifiers. The IPs are mostly created for TSMC 0.18 and TSMC 0.13 technologies. The above said circuits are optimized for area, power consumption and Unity Gain Bandwidth. This tool only supports

topology specific circuits and thus lacks generality. However, it increases the flexibility and level of reuse compared to traditional approaches.

ANASIFT [113] is a company that uses equation-based approach for sizing of analog circuits using symbolic equations. It provides different tools such as, ANASCOPE, AMPSO, AMPSO-OADFM and AASPICE for analog symbolic analysis, high performance analog optimization, analog design for manufacturability and analog simulation, respectively. These tools result in an optimized SPICE netlist as the output.

2.3 Summary

The knowledge-based optimization is used in one of the early design tools that determines the design parameters using circuit specific design plan. It can design all circuits for which design plan can be derived after which the execution time is reduced. However, this approach suffers from many drawbacks. One of them is creation of design plan for every topology which is a tedious task and requires the knowledge of skilled designers. Also, the design plans needs an update with change in process technology resulting in the creation of large library which is both time consuming and costly. The accuracy of knowledge-based approach is also limited considering the design of complex circuits in lower technology. Moreover, creation of topology specific design plan limits the generality of the design process.

Optimization based approaches overcome the disadvantages of knowledge-based approaches featuring higher level of generality and better accuracy. However, the parameters that decide the efficiency of the optimization-based design methodology are preparatory effort, execution time, generality, complexity and accuracy. In simulation-based design methodology, the circuit simulator is used within the optimization loop easing the measurement of circuit performance resulting in shorter setup time. Whereas, in equation-based design methodology, the evaluation of the circuit performance is done using a set of simple design equations which clearly effects the accuracy and thus the resulting performance. Symbolic analyzers can be used to generate the equations for evaluating various performance metrics. However, including new performance metrics into the symbolic analyser is again a time-consuming task. Moreover, the neural network based optimization results in comparatively higher preparatory effort in the process of generation of training samples using circuit simulator. The accuracy of the design can be improved by increasing the training samples.

The execution time is another important factor that effects the effectiveness of the approach used. The simulation-based approach calls a circuit simulator each time an optimization loop is executed which makes it costly in terms of computation time. Equation-based approach uses symbolic equations for evaluation which reduces the execution time when compared to simulation-based approach. Neural network-based approach provides fastest execution of all approaches due to the simplified model used.

The generality of the tool is based on its ability to design number of circuits without much human intervention. Whereas, complexity deals with the number of devices in a circuit that can be designed automatically besides the number of performance metrics considered. The simulation-based approach results in high generality as the designer can choose to define custom performance metrics by extending the test bench. This increases its ability to increase number of circuits the tool can handle. The neural network-based approach also results in similar generality due to its similarity with simulation-based approach. However, the generality of equation-based approach depends on its ability to evaluate the performance metrics using symbolic analyzers. Changing device model is practically more difficult in equation-based approach, whereas simulation-based can be adapted to the changes by updating the models in the circuit simulators like HSPICE or Spectre. Similarly, the neural network-based approach can handle varied range of circuits at the cost of large set of training samples. For equation-based approach, the size of the equations is directly proportional to the number of design parameters and hence requires large memory resources. Simulation-based approach suffers from high CPU time due to continual calls to circuit simulator. However, high speed computing resources reduce this time while increasing the efficiency.

Chapter 3

Design Optimization of Analog Circuits Using GWO and WOA Algorithms

3.1 Introduction

Analog circuits form one of the prominent blocks in modern electronic systems that serve as an interface between the signals from real world and digital realm. The importance of analog circuits cannot be neglected as the analog circuits in ICs imposes a major restriction on the design performance and overall cost. The automation of digital circuits has become a successful attempt as a result of the research since few decades, but analog circuit automation is challenging due to perplexed design. This chapter deals with the second phase of the circuit design process, i.e., circuit sizing of CMOS analog circuits using equation-based design methodology with different conventional metaheuristic algorithms. For the validation of the performance of the presented optimization technique, two of the most commonly used analog circuits i.e., CMOS DIFFAMP and two-stage CMOS OPAMP, are designed for optimum MOS transistor area.

In literature, different heuristic algorithms were applied for analog circuit design problem. However, considering the effectiveness of Grey Wolf Optimization (GWO) [114] and Whale Optimization Algorithms (WOA) [115] in comparison to state-of-the art algorithms, they are applied to analog circuit design problems. This chapter starts with the overview of GWO and WOA algorithms followed by formulation of cost function and circuit sizing. Moreover, the comprehensive and demonstrative results and their validation is also presented.

3.2 Grey wolf optimization algorithm [114]

Grey wolves belong to Canidae family which are considered as apex predators, prefer to live in a pack of 5-12 on an average. A pack has a male and female as leaders known as alpha (α) and are responsible for making important decisions. The next level of grey wolves is beta (β) that are subordinates to the alpha in making decisions which plays the role of advisor in a pack. The least level grey wolf is omega (ω) which always have to stand up to all other dominant (α, β) wolves. Delta (δ) wolves are another class of wolves which are neither of any type of wolves which play the role of scouts, hunters, elders and caretakers.

Another interesting social behavior of grey wolves besides the social hierarchy of wolves is group hunting, which starts with tracking, chasing and approaching the prey. Then the prey is encircled and pestered till its movement is stopped and then finishes by attacking it. The GWO algorithm is proposed by Syedali Mirjalili et. al., which is outlined based on the above-said behavior and discusses the mathematical model outlining the social hierarchy and hunting behavior of grey wolfs.

$$D_j = |C_j X_{pj}(t) - X_j(t)| \tag{3.1}$$

$$X_j(t+1) = X_{pj}(t) - A_j D_j \tag{3.2}$$

where, 'A_j' and 'C_j' are coefficient vectors, 'D_j' is the distance between the prey and current search agent, '$X_{pj}(t)$' and '$X_j(t)$' are position vectors of prey and grey wolf at the current iteration '(t)', respectively, || is the absolute value [115] and 'j' is the search agent number. The calculation of vectors 'A_j' and 'C_j' is as follows,

$$A_j = 2\,A_j\,r_{1j} - A_j \tag{3.3}$$

$$C_j = 2\,r_{2j} \tag{3.4}$$

where 'r_1' and 'r_2' are the random vectors within range [0, 1] and 'A_j' has its components decreased linearly from 2 to 0 during the iterations. Assuming a two dimentional position vector with prey position (X_p, Y_p) and the grey wolf at position (X, Y), the current position of the grey wolf is updated while encircling the prey with respect to the position of prey. The values of vectors 'A_j' and 'C_j' are adjusted to reach different positions around the prey. Similar position update concept can be extended in 3 dimensional and n dimensional domain, where the motion of the wolves is in the form of hyperspheres with best solution obtained at the center. Alpha wolves usually guide the hunt, with beta and delta following them. To obtain mathematical simulation of hunting behavior, we have alpha, beta and delta which are supposed to have the

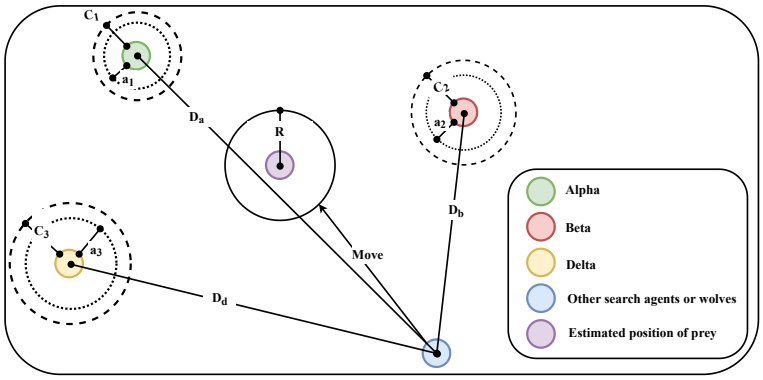


Figure 3.1: Position update of search agents in GWO.

information of potential prey location. Here, the first three best solutions obtained are saved as alpha, beta and delta while other search agents are assisted by these best solutions in updating their positions.

$$D_{aj} = \left|C_{1j}\, X_{aj} - X_j\right|,\ D_{bj} = \left|C_{2j}\, X_{bj} - X_j\right|,\ D_{dj} = \left|C_{3j}\, X_{dj} - X_j\right| \tag{3.5}$$

$$X_{1j} = \left|X_{ja} - A_{1j}\left(D_{aj}\right)\right|,\ X_{2j} = \left|X_{bj} - A_{2j}\left(D_{bj}\right)\right|,\ X_{3j} = \left|X_{jd} - A_{3j}\left(D_{dj}\right)\right| \tag{3.6}$$

$$X_j\,(t+1) = \frac{\left(X_{1j} + X_{2j} + X_{3j}\right)}{3} \tag{3.7}$$

Figure 3.1 shows the position update of search agents in accordance with alpha, beta and delta in a search space. Alpha, beta and delta defines a circle in the search space within which the final position is present.

Optimization algorithm operates in two phases i.e., exploitation and exploration. In GWO, the exploitation is done by reducing the value of 'A_j' from 2 to 0 during the iterations and 'A_j' takes random value within the interval $[-2a, 2a]$. The encircling mechanism here shows the exploration but it still needs additional operators to highlight exploration. Whereas, another vector component that helps in exploration is 'C_j' which considers any random value within the range [0, 2]. Algorithm 1 shows the pseudo code for the grey wolf optimization algorithm which starts with initialization of population, and vectors 'a_j', 'A_j' and 'C_j' followed by iterative updating of positions as shown in Equations 3.1. The vectors 'a_j', 'A_j' and 'C_j' are updated using Equations 3.3 and 3.4 and positions of alpha, beta and delta are updated with respect to best solution. Finally, the best solution is returned when the termination criteria is satisfied.

The best solution obtained so far is saved based on the social hierarchy over the course of iterations. The adaptive values of 'a_j' and 'A_j' helps in smooth transition between exploitation

Algorithm 1 Grey wolf optimization algorithm.

Initialize the grey wolf population X_i (i = 1, 2, 3, 4, ... , n).
Initialize a_j, A_j,andC_j.
Calculate the fitness of each search agent.
X_a = the best search agent.
X_b = the second best search agent.
X_d = the third best search agent.
while (t< Maximum number of iterations)
for each search agent
Update the position of the current search agent.
end for
Update a_j, A_j,andC_j.
Calculate the fitness of all search agents.
Update X_a, X_bandX_d.
$t = t + 1$
end while
return X_a

and exploration in GWO. As the value of 'A_j' is decreased, the number of iterations are divided equally and each half is dedicated to exploitation ($|A| < 1$) and exploration ($|A| \geq 1$).

3.3 Whale optimization algorithm [115]

The WOA algorithm is proposed by Syedali Mirjalili et. al. based on the mathematical model derived from the social behavior observed in humpback whales that deals with foraging and hunting their favorite prey such as, small fish and krill herds. The special hunting method observed in humpback whales is bubble-net feeding method which starts with diving into ocean up to 12 meters deep and creating bubbles with spiral and upward movement towards the prey. The following maneuver includes coral loop, lobtail and capture loop which are explained in [116]. The mathematical modeling, for optimization, is done based on the spiral bubble-net maneuver in humpback whales.

The hunting process or exploitation phase starts with encircling the prey after recognizing its position as a target or close to optimum solution. After defining the best search agent, search agents update their positions towards the best solution, which is mathematically represented as follows:

$$D_j = \left| C_j \, X_{jp}(t) - X_j(t) \right| \tag{3.8}$$

$$X_j(t+1) = \left| X_{jp}(t) - A_j \, D_j \right| \tag{3.9}$$

where 't' is current iteration, 'X_{pj}' is the best solution position vector obtained, 'X_j' is position vector, 'A_j' and C_j are coefficient vectors. The position 'X_{pj}' must be updated in each iteration to obtain better solution. The vectors 'A_j' and 'C_j' are calculated using following equations:

$$A_j = 2\,A_j\,r_{1j} - A_j \tag{3.10}$$

$$C_j = 2\,r_{2j} \tag{3.11}$$

where 'r_1' ,'r_2' are random vectors within range [0, 1], 'A_j' has its components decreased linearly from 2 to 0 during the iterations and 'j' is the search agent number.

The mathematical modeling of the bubble-net behavior is done based on two approaches i.e., shrinking circles mechanism and spiral position update. The behavior of shrinking circles mechanism is obtained by reducing the value of 'A_j' as shown in Equation (3.10) which eventually reduces the fluctuation range of 'A_j' i.e., 'A_j' is a random value in a range $[-a, a]$. If the value of 'A_j' is set randomly between [-1, 1], the new position is anywhere between current best position and original position of an agent. The spiral position update firstly calculates the distance between whale and prey. Then, the equation is modelled to mimic the helix-shaped movement of the whale as shown below:

$$X_j\,(t+1\,) = D_j^{'} e^{bl} \cos\,(2\pi l\,) + X_{pj}\,(t) \tag{3.12}$$

where '$D_j = |X_{pj}(t) - X_j(t)|$' is the distance between the prey and i^{th} whale, b is the logarithmic spiral defining constant and 'l' takes any random number in range [-1,1]. Both of these phenomena are merged to depict the behavior of the humpback whales which is modelled as follows:

$$X_j\,(t+1) = \begin{cases} \left|X_{jp}\,(t) - A_j\,D_j\right| & ;if\,p < 0.5 \\ \\ D_j^{'} e^{bl} \cos\,(2\pi l\,) + X_{pj}\,(t) & ;if\,p \geq 0.5 \end{cases} \tag{3.13}$$

where, 'p' is the random value ranging between 0 and 1 which divides the search agents equally for encircling and spiral movement around the prey. Besides bubble-net method, the humpback whales follow random search, according to each other's position, for searching prey. This process depends on the value of 'A_j' i.e., when 'A_j' is less than -1 or greater than 1, the search agent is forced to move away from the reference whale. In the exploration phase, the position update is done according to the randomly chosen search agent instead of best search agent so far. The exploration phase is mathematically modeled as follows:

$$D_j = |C_j X_{rj}(t) - X_j(t)| \tag{3.14}$$

$$X_j(t+1) = |X_{rj}(t) - A_j D_j| \tag{3.15}$$

where (X_{rj}) is a random search agent from current population. The pseudo-code for the WOA is shown in Algorithm 2.

Algorithm 2 Whale optimization algorithm.

Initialize the whale population X_i (i = 1, 2, 3, 4, ... , n).
Calculate the fitness of each search agent.
X_p = the best search agent.
while (t< Maximum number of iterations)
for each search agent
Update $a_j, A_j, C_j, l$and p.
if ($p < 0.5$)
if($|A| < 1$)
Update the position of the current search agent using Equation 3.9.
else if ($|A| \geq 1$)
Select a random search agent (X_r).
Update the position of the current search agent using Equation 3.15.
end if
else if ($p \geq 0.5$)
Update the position of the current search agent using Equation 3.13.
end if
end for
Check if any search agent goes beyond the search space and amend it.
Calculate the fitness of all search agents.
Update X_p if there is a better solution.
$t = t + 1$
end while
return X_p

3.4 Formulation of cost function

The design of CMOS DIFFAMP and two-stage CMOS OPAMP, using GWO and WOA as the optimization algorithms is presented. The design specifications considered for designing the above-said amplifier circuits are voltage gain (A_v), unity gain bandwidth (UGB)/ cutoff frequency (f_{-3dB}), power dissipation (P_d), input common mode range ($ICMR-$ and $ICMR+$), and slew rate (SR). Besides design specifications, design parameters i.e., aspect ratios (W/L) of all the transistors in circuits and bias current (I_{bias}), which are crucial in designing an analog circuit are considered. The schematics for CMOS DIFFAMP and two-stage CMOS OPAMP are shown in Figures 3.2 and 3.3, respectively. The variables that are initialized before starting the iterative optimization process include: power supply (V_{dd}), threshold voltage inputs for

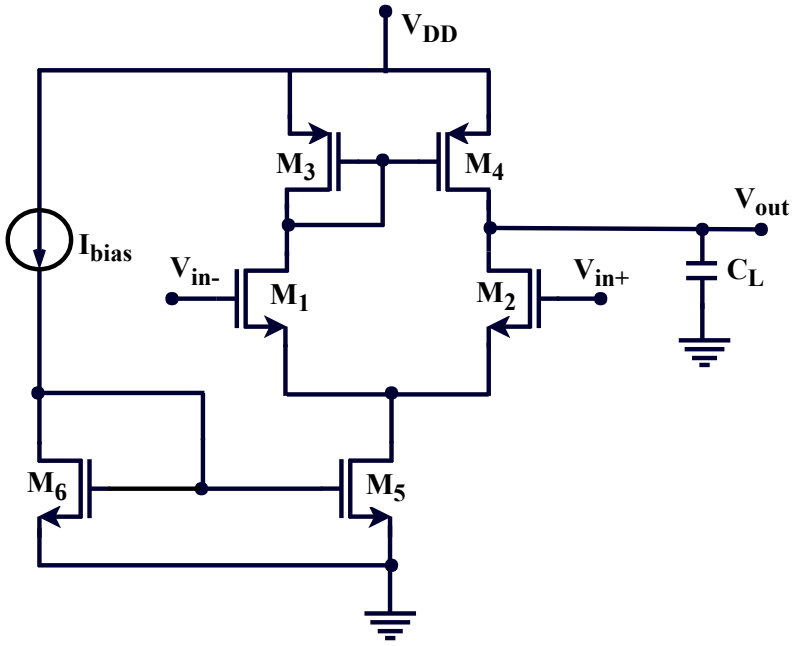


Figure 3.2: Schematic of CMOS differential amplifier.

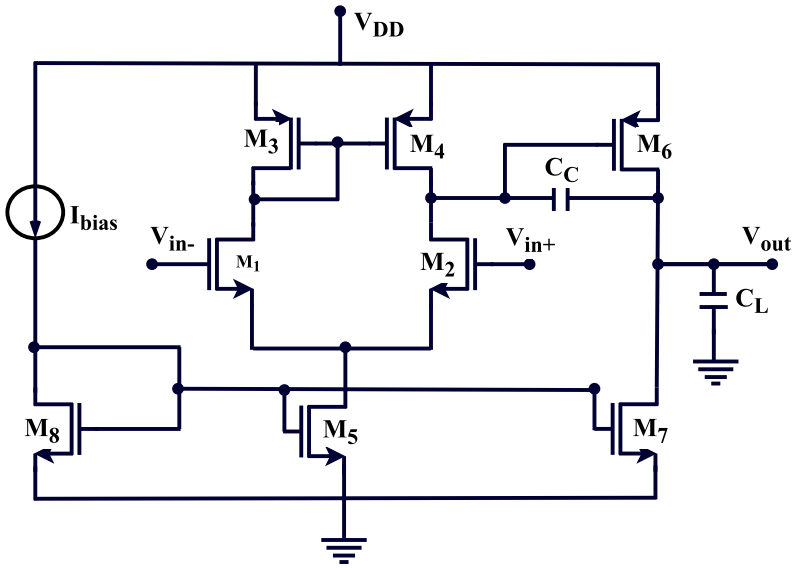


Figure 3.3: Schematic of two-stage CMOS OPAMP.

NMOS and PMOS (V_{tn} and V_{tp}), gate to source voltage (V_{GS}) of MOS transistor, drain to source voltage (V_{DS}) of MOSFET, transconductance parameter of PMOS and NMOS ($K_p^{'} = \mu_p C_{ox}$ and $K_n^{'} = \mu_n C_{ox}$), mobility of charge carriers μ_n for electrons and μ_p for holes), gate oxide capacitance (C_{ox}), channel length modulation parameters (λ_p for PMOS and λ_n for NMOS), transconductance (g_m), output conductance (g_{ds}), output resistance (R_{out}) and drain current (I_D). The design of CMOS DIFFAMP and CMOS two-stage OPAMP is done using the methodology shown in Appendix A.1 [117].

The initial population size for the algorithms is considered to be a matrix of size [P × Q], where P = 60 and Q = 7, where P is number of particles and Q is the particle vector. The particle vectors for the CMOS DIFFAMP and two-stage CMOS OPAMP circuits are as follows:

$$X_{diffamp} = [A_v, C_L, SR, ICMR-, ICMR+, f_{-3dB}, P_d] \tag{3.16}$$

$$X_{opamp} = [A_v, C_L, SR, ICMR-, ICMR+, UGB, P_d] \tag{3.17}$$

Cost function or fitness function is a mathematical function that evaluates the design requirements to obtain its minimum (or maximum) value using design variables. Here, the cost function is derived to achieve the optimum aspect ratios of all transistors in the circuit so as to reduce the total MOS area, using the relationships between different parameters used in the design. The population is computed by fitness of each vector. Usually, many parameters can be optimized in analog circuit optimization tasks. Hence, cost function is necessary for determining the fitness of the circuit. Here, cost function is characterized as the total MOS area occupied (sum of widths × lengths) by all the transistors because the objective that is considered for optimization (or minimization) is area occupied by the circuit. Therefore, the Cost Function (CF) is given by

$$CF = \sum_{i=1}^{N} \left(S_i \times L_i^2 \right) \tag{3.18}$$

where N is the total number of transistors in a circuit with the desired value of the cost function to be minimum for given circuits. Hence, these algorithms are utilized for obtaining the optimal value of the cost function individually.

3.5 Simulation Results

The algorithms, i.e., GWO and WOA, are used to find the optimal aspect ratios of all the transistors using MATLAB 2017a on CPU Intel core (TM) i7-4790 at 3.60 GHz. These amplifiers are implemented using MATLAB and then the aspect ratios obtained are used to simulate the amplifiers in CADENCE analog design environment. The technology parameters are chosen from 180nm CMOS standard process. The constants considered for designing above discussed amplifiers are V_{dd} = 1.8 V, V_{tp} = 0.4523 V, V_{tn} = 0.3215 V, K_p = 80.6 $\mu V/A^2$ and K_n = 351.56 $\mu V/A^2$. The widths and lengths of the transistors are considered to be in the range (0.5 μm, 10 μm). The length of transistors is kept high for designing amplifiers to suppress the channel length modulation effects. The design specifications for CMOS differential amplifier and two-stage operational amplifier are illustrated in column 2 of Tables 3.3 and 3.4, respectively. The

Table 3.1: Design parameters of CMOS differential amplifier using GWO, WOA, GSA and PSO algorithms.

Design Parameter	Differential Amplifier			
	GWO	WOA	GSA	PSO
I_b (μA)	12	12	20	21
$W_{1,2}$ (μm)	5.25	7.25	8	10.65
$W_{3,4}$ (μm)	0.75	1.55	3	2.41
W_5 (μm)	3.19	5.27	3.75	4
W_6 (μm)	2.28	3.12	3.75	2.5
L (μm)	0.75	0.75	0.75	0.75

design parameters, including the aspect ratios of transistor and bias current (I_b), attained after the optimization of CMOS amplifiers using different algorithms are illustrated in Table 3.1 and Table 3.2.

Table 3.2: Design parameters of two stage operational amplifier using GWO and WOA algorithms.

Design parameters	Simulation-based		Equation-based					
	GSA [118]	AGSA [118]	PSO [119]	WOA [120]	PSO [121]	DE [121]	GWO	WOA
W_1/L_1	4/2	4/2	7.74/0.18	4/2	4/2	4/2	10/0.54	17.54/0.54
W_3/L_3	4/2	4/2	14.4/0.18	5/2	4/2	4/2	0.5/0.54	0.75/0.54
W_5/L_5	4/2	4/2	1.96/0.18	2/2	4/2	2/2	18/0.54	2.65/0.54
W_6/L_6	21.94/2	7.16/2	98.23/0.18	21.54/2	33.4/2	21.5/2	2.05/0.54	6.2/0.54
W_7/L_7	11.36/2	4/2	11.97/0.18	5.38/2	16.7/2	5.38/2	15/0.54	16.5/0.54
W_8/L_8	4/2	4/2	1.96/0.18	2/2	16.7/2	5.38/2	0.95/.054	0.95/0.54
I_{bias} (μA)	45.28	30	34.46	49.0	21	49	13	13
C_C (pF)	4.4	2.2	–	2.45	2.45	2.45	1.8	1.8

Units of W_i and L_i are μm, where i = 1 to 8

Tables 3.3 and 3.4 summarize the results obtained after circuit level implementation in CADENCE IC616 with the aspect ratios shown in Table 3.1 and Table 3.2, for CMOS differential amplifier and two-stage operational amplifier, respectively. These results are also compared with those from the recent literature using state-of-the-art algorithms. It is observed from the comparison that the performance of GWO is better than other algorithms for its application in optimizing analog circuits resulting in optimum MOS area of 11.10 μm^2 and 28.02 μm^2 for CMOS differential amplifier and two-stage operational amplifier, respectively. However, besides optimizing the area, GWO and WOA algorithms result in high gain, low power consumption and high CMRR values when compared to other competing algorithms. For a fair comparison between recent techniques using algorithms such as PSO, WOA, DE, GSA, AGSA,

Table 3.3: Performance comparison of differential amplifier.

Design criteria	Perf. spec.	GSA	PSO	GWO	WOA
Technology *nm*		180 CMOS	180 CMOS	180 CMOS	180 CMOS
SR ($V/\mu s$)	≥ 10	19.59	17.92	39.74	14.79
Pd (μW)	≤ 200	71.62	92.45	73.06	70.20
PM (°)	≥ 45	89.44	87.27	87.75	89.54
f_{-3dB} (KHz)	≥ 200	91.2	155.9	457	154
Gain (dB)	≥ 40	45.07	44.06	45.80	46.06
CMRR (dB)	≥ 60	82.66	78.5	87.35	85.49
PSRR (dB)	≥ 70	90.09	87.95	78.46	92.12
MOS area (μm^2)	Min.	22.12	25.57	11.01	19.83

Table 3.4: Results obtained using GWO, WOA and its comparison using different algorithms.

Design specifications	Tar.	PSO [119]	WOA [120]	PSO [120]	DE [120]	GSA [118]	AGSA [118]	GWO	WOA
A_v (dB)	>60	59.19	74.08	61	74.08	60.14	81.13	78.35	80.13
GBW (MHz)	>3	3	3	3.85	3	3.136	3.29	15.97	4.29
PM ($degrees$)	>45	63.53	–	–	–	47.53	57.91	61.27	62.66
SR ($V/\mu s$)	>10	18.35	10	10	20	10.29	12.34	16.62	13.44
Pd (mW)	<2.5	0.18	1.14	0.98	2.5	1.05	0.332	0.16	0.266
CL (pF)	>7	–	7	7	7	10.02	10	7	7
V_{ICmin} (V)	>0.3	–	-0.047	-0.01	-0.01	-0.86	-1.22	0.38	0.45
V_{ICmax} (V)	<1.6	–	1.1	1.1	1.11	1.8	1.79	1.22	1.35
$CMRR$ (dB)	>60	67.08	–	–	–	81.99	84.6	92.76	92.12
$PSRR+$ (dB)	>70	63.84	–	–	–	77.36	97.45	78.27	80.07
$PSRR-$ (dB)	>70	99.16	–	–	–	86.82	84.86	72.25	75.54
$Area(\mu m^2)$	Obj.	28.52	93.86	148.2	93.8	114.6	70.32	28.02	33.44
FOM_1	Max.	–	0.43	1.283	0.43	0.69	1.09	8.6	2.31
FOM_2	Max.	–	0.197	0.186	0.01	0.26	1.41	24.32	3.38
Tech. (μm)		0.18	0.18	0.18	0.18	0.35	0.35	0.18	0.18

and GWO, with application to circuit sizing that utilize different technologies, two Figures Of Merit (FOM) [122], [123] are considered as shown in Equation 3.19 an Equation 3.20. These FOMs demonstrate that the performance of GWO algorithm is superior to other competing algorithms.

$$FOM_1 = \frac{C_L\,(pF)\,UGB\,(MHz)}{I_{bias}\,(\mu A)} \tag{3.19}$$

$$FOM_2 = \frac{C_L\,(pF)\,UGB\,(MHz)}{P_d\,(\mu W)\,Area(\mu m)^2 10^{12}} \tag{3.20}$$

3.6 Summary

Automation of analog CMOS circuits is one of the difficult and time consuming tasks. An evolutionary optimization technique based on GWO and WOA algorithms is used to optimize analog amplifier circuits i.e., CMOS DIFFAMP and two-stage CMOS OPAMP, aiming to meet the design specifications such as slew rate, dc gain, phase margin, PSRR, CMRR and power consumption. The optimization process is performed in MATLAB, and the results are validated using CADENCE. The approach of using GWO and WOA algorithm in analog circuit sizing is proved to be better than related algorithms due to faster convergence to minimum area (optimum solution). However, as the equation-based design methodology uses simple design equations to evaluate the performance metrics, there is a need for fine tuning to obtain desired performance.

Chapter 4

EGWO Algorithm for Analog Circuit Optimization

4.1 Introduction

With the scaling of CMOS process, the demand for the integration of both analog and digital circuits on the same die has increased. Though the analog circuitry in an integrated circuit (IC) occupies less portion when compared to its digital counterpart, its complexity and non-linearity makes the design process more challenging. The technology scaling has put more constraints on the analog circuit design, therefore making the design process more complicated, time-consuming, skillful and costly, resulting in increased overall time-to-market. The evolution of reliable tools for analog circuit automation is not mature enough, therefore making it an emerging field of research.

A novel circuit sizing technique with improved accuracy and efficiency is proposed to resolve the sizing issues in the analog circuit design. The GWO algorithm has the total number of iterations divided equally for exploration and exploitation, overlooking at the impact of balance between these two phases, aimed for the convergence at globally optimal solution. An enhanced version of a typical GWO algorithm termed as Enhanced Grey Wolf Optimization (EGWO) algorithm are presented with improved exploration ability and is successfully applied in analog circuit design. A set of 23 classical benchmark functions is evaluated and the outcomes are compared with recent state-of-the-art algorithms. The CMOS DIFFAMP and two-stage CMOS OPAMP circuit, realized in 180 nm CMOS standard process, are used as a benchmarks to vali-

date the efficiency of the proposed optimization technique. A statistical study is also conducted over the final solution to investigate the exploration ability of the algorithm proving it to be one of the robust and reliable techniques.

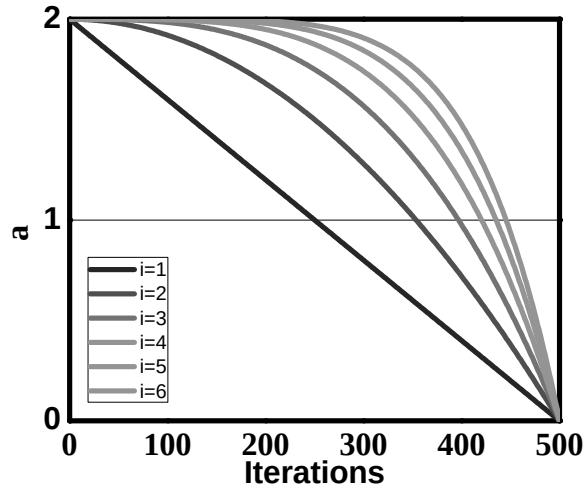
4.2 Enhanced Grey Wolf Optimization Algorithm

The key purpose of an optimization algorithm is to obtain a global optimum solution. In this process, it has to accomplish two phases i.e., exploration and exploitation. Exploration deals with scattering of search agents throughout the search space, followed by converging towards global optimum in the exploitation phase. Proper balance needs to be maintained between exploration and exploitation in order to obtain convergence at global optimum by avoiding local minima. According to the No Free Lunch (NFL) theorem [124], there exists no single algorithm that can solve all the optimization problems due to the trade-offs between exploration and exploitation. Hence, there is always a need for more sophisticated algorithms to tackle different optimization problems. An effort to refine one of the existing algorithms is presented in the form of EGWO.

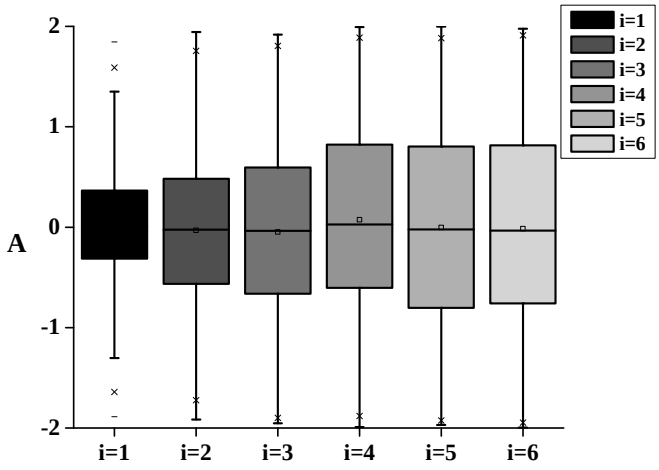
The GWO algorithm discusses the mathematical model outlining the social hierarchy and hunting behavior of grey wolfs. In GWO, the exploration and exploitation depend on two parameters i.e., 'a' and 'A_j'. Half of the iterations are devoted to exploration phase i.e., when '$\left|A_j\right| \geq 1$' and the other half are assigned to exploitation phase i.e., when '$\left|A_j\right| < 1$', where 'j' is the search agent number. Higher exploration results in lower probability of stagnation at local optimum. The modifications are done keeping in mind the ratio of exploration and exploitation that is to be maintained. The EGWO improves the exploration ability by increasing the non-linearity of parameter 'A_j' as shown below.

$$a = \begin{cases} 2\left(1 - \frac{t}{T}\right) & ; for\ conventional\ GWO \\ \\ 2\left(1 - \frac{t^i}{T^i}\right) & ; for\ enhanced\ GWO \end{cases} \tag{4.1}$$

where, t is the present iteration, T is the maximum number of iterations. The variation of 'A_j' over the course of 500 iterations for different values of i is shown in Figure 4.1a that helps in determining the value of i (= 3) for slightly increasing the number of iterations assigned for exploration. When $i = 1$, the search agents are divided equally between exploration and exploitation phases. As the value of i increases, the iterations assigned for exploration are higher when compared to those assigned for exploration making the relation a nonlinear one.



(a)



(b)

Figure 4.1: (a)Variation of a over iterations, (b) Box plot for values of A.

This helps in improving the exploration ability. Figure 4.1b shows the variation of 'A_j' as an effect of variation in 'A_j' using box plot . Besides changing the value of 'A_j', in order to improve the exploration ability further, the procedure to update the position is modified by incorporating slight randomness in the position update of search agents. Here, a strategy of calculating the vectors 'D'_{aj}', 'D'_{bj}' and 'D'_{dj}' is applied to avoid trapping at local optima. The formulation of updated positions is as follows:

$$D'_{aj} = \left|C_{1j}\, X_{raj} - X_{rbj}\right|,\ D'_{bj} = \left|C_{2j}\, X_{rbj} - X_{rdj}\right|,\ D'_{dj} = \left|C_{3j}\, X_{rdj} - X_{raj}\right| \tag{4.2}$$

$$X'_{1j} = \left|X_{aj} - A_{1j}\left(D'_{aj}\right)\right|,\ X'_{2j} = \left|X_{bj} - A_{2j}\left(D'_{bj}\right)\right|,\ X'_{3j} = \left|X_{dj} - A_{3j}\left(D'_{dj}\right)\right| \tag{4.3}$$

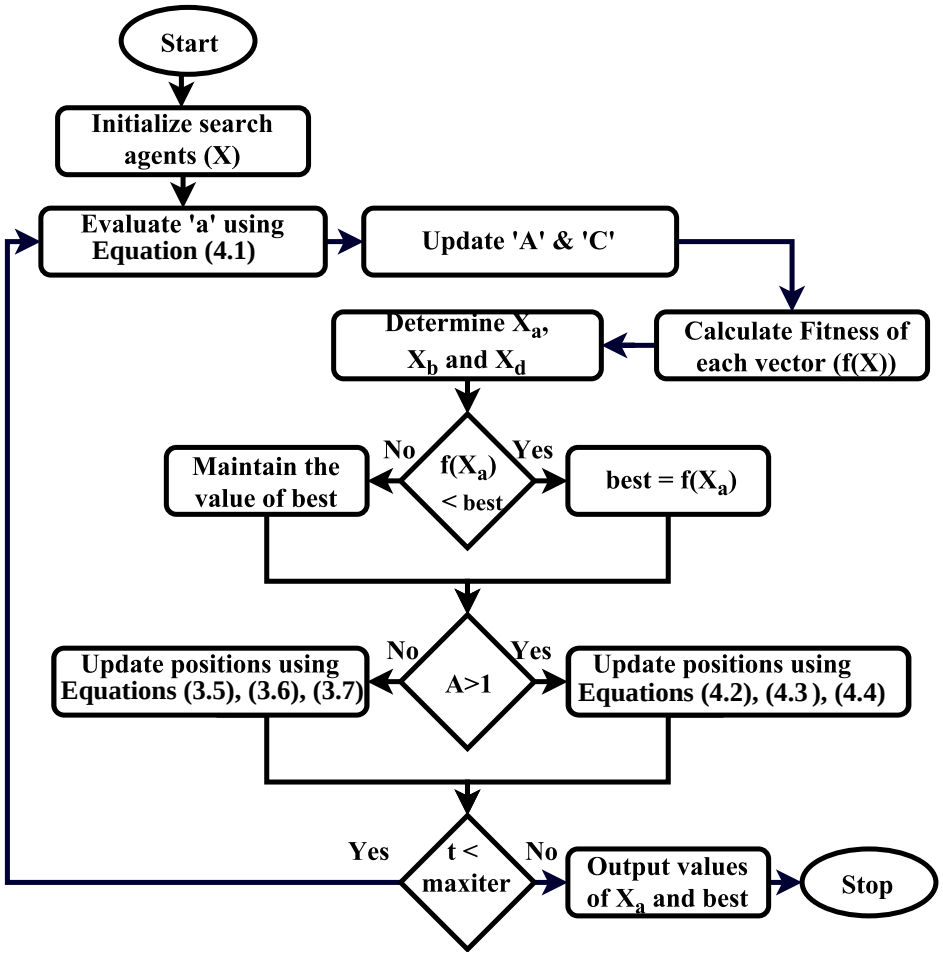


Figure 4.2: Optimization flow for EGWO.

$$X_j^{'}(t+1) = \frac{\left(X_{1j}^{'} + X_{2j}^{'} + X_{3j}^{'}\right)}{3} \tag{4.4}$$

where, '*ra*', '*rb*' and '*rd*' are random search agents from within the population such that $ra \neq rb \neq rd$ and '*j*' is the search agent number. The vectors '$D_{aj}^{'}$', '$D_{bj}^{'}$' and '$D_{dj}^{'}$' are used only when the value of A is greater than 1. The optimization framework, shown in Figure 4.2, demonstrates the iterative process of optimization using EGWO.

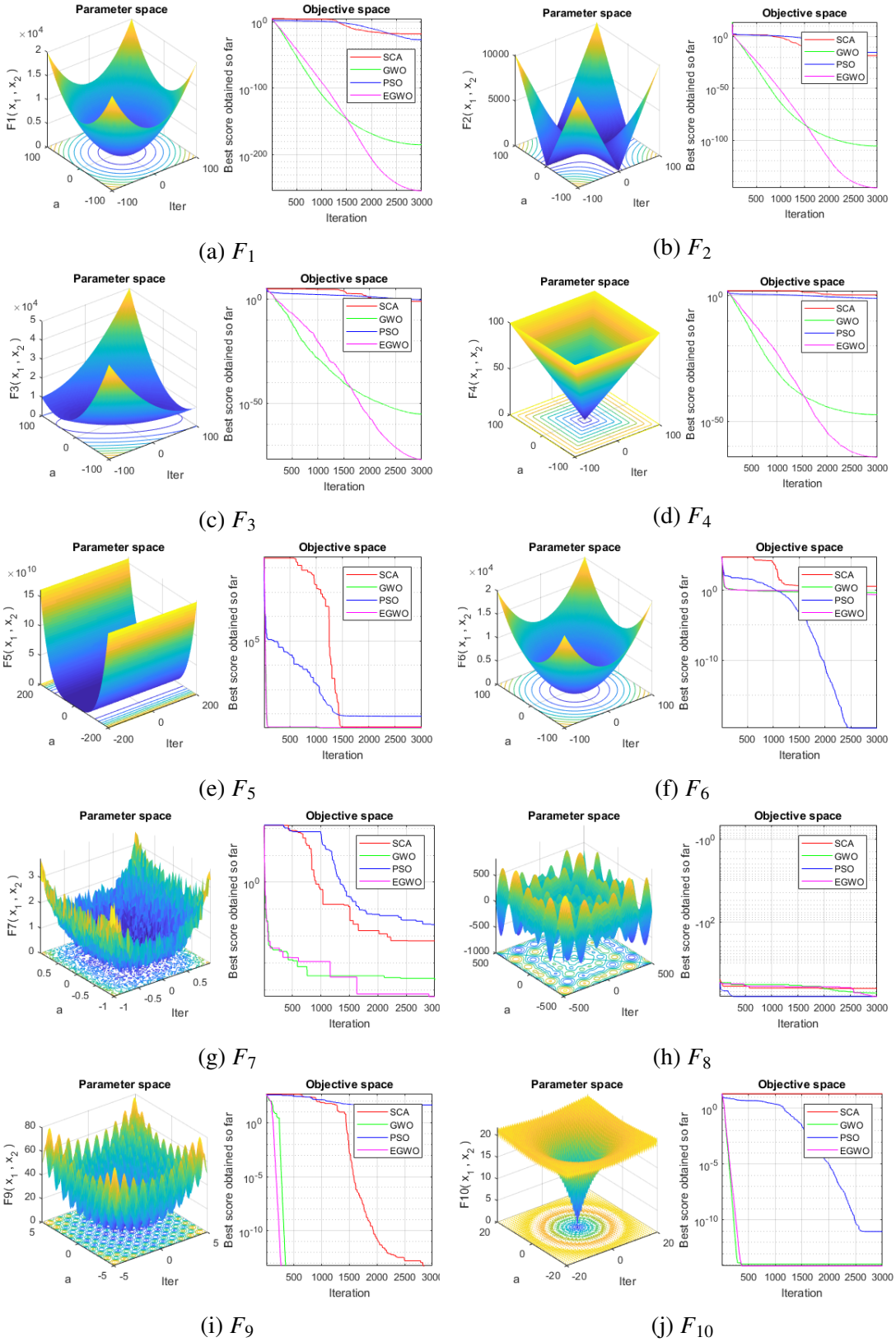
4.3 Performance evaluation of EGWO

The performance of the Enhanced GWO is compared with other algorithms with respect to 23 classical and popular benchmark functions A.2, employed by many researchers, to check the efficiency of the proposed algorithm.

Table 4.1: Minimization results of benchmark functions over 20 independent runs for $F_1 - F_{23}$.

Function		EGWO	mGWO	GWO	SCA	PSO
F_1	Mean	**6.39E-255**	1.77E-203	7.27E-185	6.87E-14	3.65E-22
	Best	**0**	3.40E-210	1.50E-188	4.58E-21	6.04E-30
	Worst	**1.24E-253**	3.32E-202	1.06E-183	7.63E-13	5.29E-21
F_2	Mean	**5.54E-146**	7.04E-119	1.23E-106	4.23E-18	3.92E-10
	Best	**4.48E-149**	5.12E-121	3.25E-108	1.64E-22	1.13E-14
	Worst	**1.02E-144**	3.82E-118	5.16E-106	2.46E-17	5.56E-09
F_3	Mean	**7.03E-69**	1.33E-52	1.00E-52	477.45	0.10697
	Best	**2.21E-79**	3.20E-66	7.93E-63	6.56457	0.03107
	Worst	**1.34E-67**	1.59E-51	1.42E-51	2016.502	0.22511
F_4	Mean	**1.78E-68**	3.84E-53	1.21E-45	2.53889	0.12472
	Best	**7.30E-71**	2.97E-56	5.60E-48	0.00605	0.0372
	Worst	**1.08E-67**	2.77E-52	7.92E-45	9.33451	0.35465
F_5	Mean	**26.2359**	26.7851	26.50665	27.94805	49.58214
	Best	25.11832	26.17388	25.23714	27.31307	**3.96877**
	Worst	27.15566	28.54891	**27.14311**	28.87265	124.1508
F_6	Mean	0.51215	0.61415	0.63433	3.94743	**9.69E-21**
	Best	1.99E-05	0.25023	1.76E-06	3.26782	**2.64E-29**
	Worst	1.00192	1.49914	1.2534	4.40688	**1.66E-19**
F_7	Mean	**0.00016**	0.00032	0.00021	0.00689	0.02041
	Best	**2.57E-05**	0.00011	6.92E-05	0.00049	0.00786
	Worst	**0.00031**	0.00076	0.00049	0.02634	0.03687
F_8	Mean	-5806.37	-5705.05	-6485.47	-4195.37	**-6535.5**
	Best	-7523.56	-6501.72	-7602.48	-4901.78	**-7634.22**
	Worst	-3962.7	-4901.58	**-5273.48**	-3651.37	-4103.55
F_9	Mean	**0**	**0**	**0**	1.46292	37.31178
	Best	**0**	**0**	**0**	**0**	20.89413
	Worst	**0**	**0**	**0**	23.17803	56.71271
F_{10}	Mean	**7.82E-15**	**7.82E-15**	8.53E-15	12.56573	1.46E-11
	Best	**4.44E-15**	**4.44E-15**	7.99E-15	1.44E-10	1.51E-14
	Worst	**7.99E-15**	**7.99E-15**	1.51E-14	20.23885	1.73E-10
F_{11}	Mean	**0**	0.00129	0.00201	0.02531	0.00824
	Best	**0**	**0**	**0**	**0**	**0**
	Worst	**0**	0.0131	0.03233	0.31675	0.04924

Function		EGWO	mGWO	GWO	SCA	PSO
F_{12}	Mean	**0.01091**	0.05028	0.03011	0.50929	0.01136
	Best	1.93E-06	0.01968	0.00655	0.35015	**1.83E-31**
	Worst	**0.04596**	0.14342	0.07216	0.91857	0.10367
F_{13}	Mean	0.39694	0.52992	0.48782	2.17183	**0.00275**
	Best	0.09811	0.19838	0.21072	1.86228	**2.95E-29**
	Worst	1.11513	0.82957	0.83777	2.47633	**0.01099**
F_{14}	Mean	5.11802	4.22526	4.91577	**1.29562**	3.02281
	Best	**0.998**	**0.998**	**0.998**	**0.998**	**0.998**
	Worst	10.76318	12.67051	12.67051	**2.9821**	6.90333
F_{15}	Mean	0.00537	0.00443	0.00532	0.00077	**0.00064**
	Best	0.00031	0.00031	0.00031	0.00032	**0.00031**
	Worst	0.02036	0.02036	0.02036	0.00145	**0.00107**
F_{16}	Mean	**-1.03163**	**-1.03163**	**-1.03163**	-1.03162	**-1.03163**
	Best	**-1.03163**	**-1.03163**	**-1.03163**	-1.03162	**-1.03163**
	Worst	**-1.03163**	**-1.03163**	**-1.03163**	-1.0316	**-1.03163**
F_{17}	Mean	0.397888	0.397888	0.397893	0.398391	**0.397887**
	Best	**0.397887**	**0.397887**	**0.397887**	0.397907	**0.397887**
	Worst	0.397894	0.397899	0.398018	0.401533	**0.397887**
F_{18}	Mean	**3**	3.000001	3.000001	3.000001	**3**
	Best	**3**	**3**	**3**	**3**	**3**
	Worst	**3**	3.000006	3.000008	3.000006	**3**
F_{19}	Mean	**-3.86266**	-3.86196	-3.86117	-3.85538	-3.86265
	Best	**-3.86278**	-3.86278	-3.86278	-3.86218	-3.86278
	Worst	**-3.8649**	-3.85592	-3.8549	-3.85418	-3.86278
F_{20}	Mean	**-3.27508**	-3.26041	-3.25905	-2.9348	-3.27444
	Best	**-3.32199**	**-3.32199**	**-3.32199**	-3.16861	**-3.32199**
	Worst	**-3.28476**	-3.08668	-3.08668	-1.91857	-3.2031
F_{21}	Mean	**-9.89793**	-9.39278	-9.6479	-3.45751	-7.62471
	Best	**-10.153199**	**-10.153199**	**-10.153199**	-7.250155	**-10.153199**
	Worst	-5.055198	-5.055197	-5.100635	-0.497293	-5.055198
F_{22}	Mean	**-10.402559**	-10.1389	-10.137125	-4.756824	-9.581963
	Best	**-10.402957**	-10.40295	-10.40292	-7.667205	-10.40294
	Worst	**-10.402101**	-5.12861	-5.087671	-0.907976	-5.087672
F_{23}	Mean	-10.535763	**-10.5361**	-10.265962	-5.282028	-9.727579
	Best	**-10.536409**	-10.5364	-10.5364	-8.559539	-10.536408
	Worst	-10.535106	**-10.5356**	-5.12848	-0.945664	-5.128481



(a) F_1 (b) F_2

(c) F_3 (d) F_4

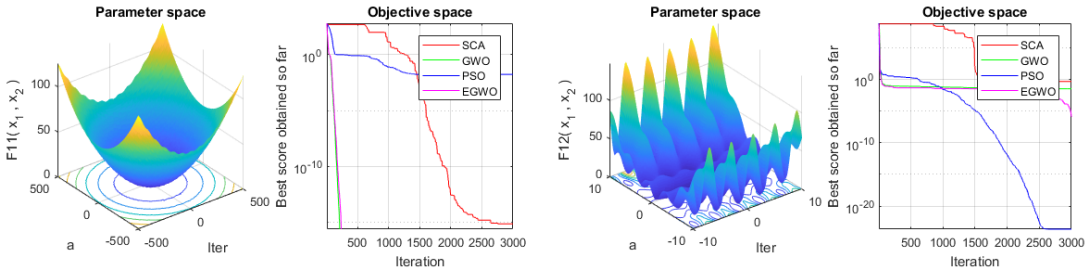
(e) F_5 (f) F_6

(g) F_7 (h) F_8

(i) F_9 (j) F_{10}

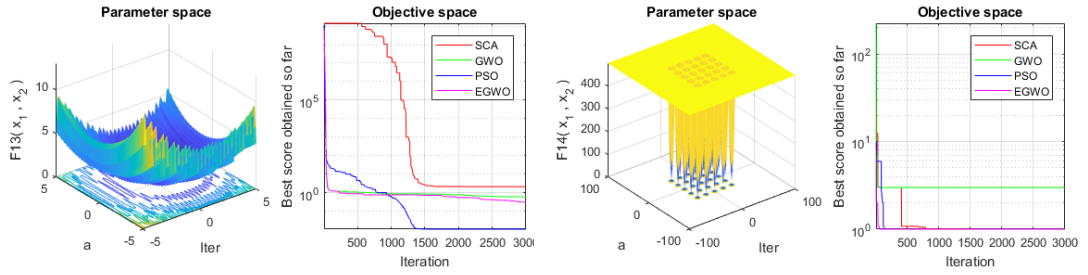
Figure 4.3: Convergence plots of benchmark functions (F_1 - F_{10}) for 3000 iterations.

These benchmark functions [125] are classified as unimodal, fixed (low) dimensional and multimodal high dimensional benchmark functions. The simulation results after evaluation of



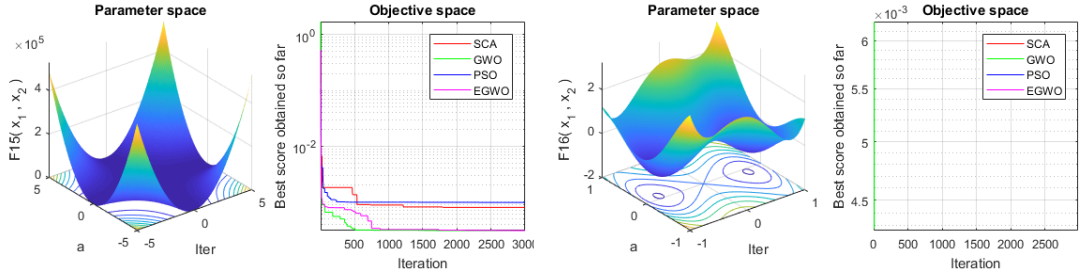
(a) F_{11}

(b) F_{12}



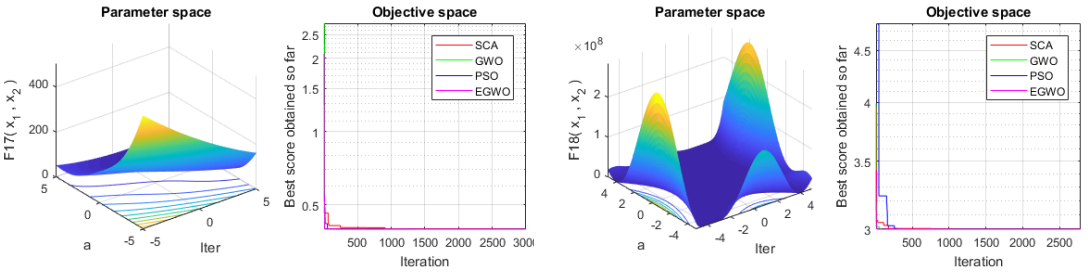
(c) F_{13}

(d) F_{14}



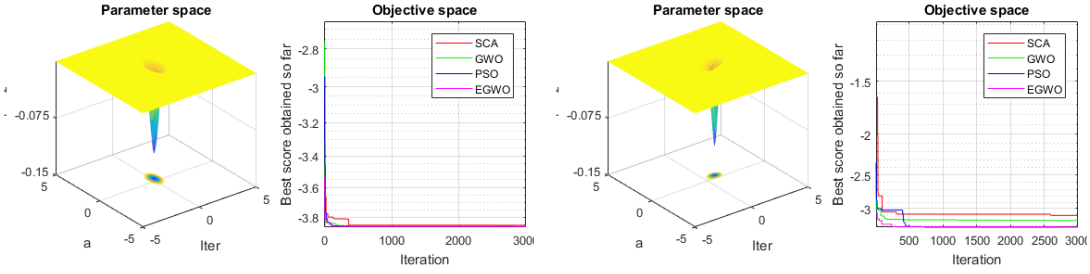
(e) F_{15}

(f) F_{16}



(g) F_{17}

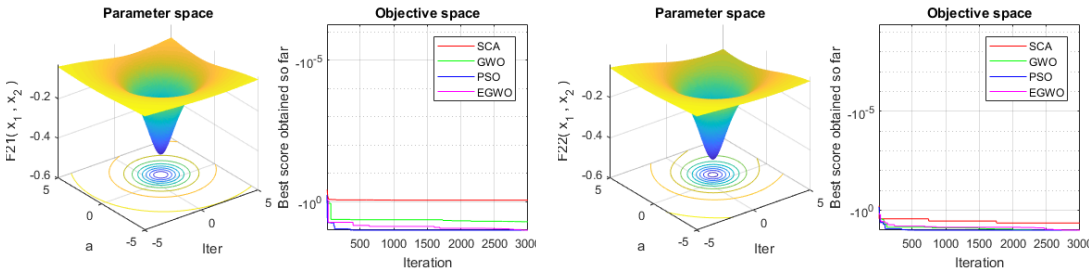
(h) F_{18}



(i) F_{19}

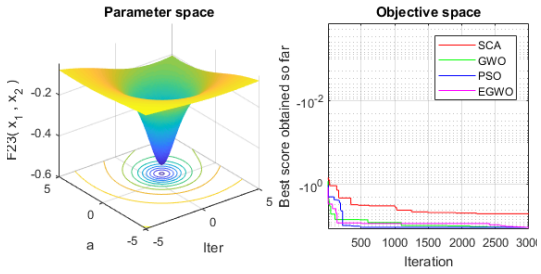
(j) F_{20}

Figure 4.4: Convergence plots of benchmark functions (F_{11} - F_{20}) for 3000 iterations.



(a) F_{21}

(b) F_{22}



(c) F_{23}

Figure 4.5: Convergence plots of benchmark functions (F_{21} - F_{23}) for 3000 iterations.

the said benchmark functions, with 20 independent runs, using proposed EGWO algorithm is compared with other algorithms namely, SCA [126], PSO [127], GWO and mGWO [128]. A statistical study shown in Table 4.1 reports the mean, worst and the best-so-far solutions after the final iteration.

Functions $F_1 - F_7$ are unimodal functions that have only global optimum solution and are used to examine optimization algorithms for the rate of convergence. The results for these unimodal functions show that EGWO outperforms the other algorithms on five out of seven for D=10. For multimodal functions $F_8 - F_{13}$ with many local minima, the final results are more important as they reflect the algorithm's ability to avoid local optima and obtain the near-global optimum. Testing on functions $F_8 - F_{13}$ show that the proposed algorithm performs better than other algorithms on three out of six multimodal high-dimensional benchmark functions. Functions $F_{14} - F_{23}$ are simpler due to a smaller number of local minima and low dimensionalities. Validation for this set of functions shows that EGWO performs better than the other algorithms on seven out of ten of the multimodal low-dimensional benchmark functions.

The convergence rates demonstrate how fast the algorithm converges to the global optimum solution over the course of iterations. The convergence rates of the EGWO, SCA, PSO, GWO and MGWO algorithms has been investigated for functions $F_1 - F_{23}$ and the same are demonstrated in Figure 4.3, Figure 4.4 & Figure 4.5. The descending trend proves the ability of the EGWO in obtaining better approximation of global optimum over the course of iterations. Overall, these results show the potential of EGWO in solving problems (of the types tested) that cannot be solved efficiently by other algorithms.

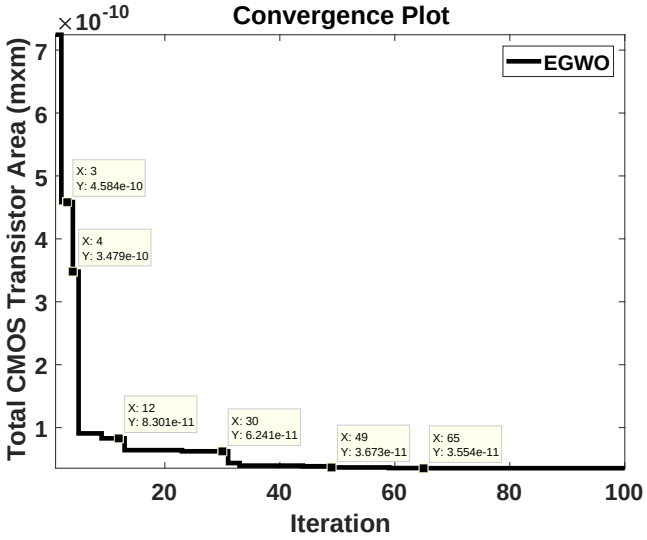


Figure 4.6: Convergence trend for optimal CMOS transistor area using EGWO.

4.4 Formulation of cost function and implementation

The circuit-level implementation and the optimization of the geometrical ratios of MOS transistors for analog circuits employing the EGWO is discussed in this section. Here, the constraints and technology parameters are initialized to obtain the design parameters i.e., aspect ratio of the transistors, bias current (I_{bias}) and capacitor values. The process starts with initialization of parameters randomly within the range specified by the designer followed by focusing on the optimization of the cost function. The algorithm results in new values of the design parameters for optimal value of cost function when the termination condition is satisfied.

The circuits used as a case study for optimization are CMOS DIFFAMP and two-stage CMOS OPAMP with miller-compensation as shown in Figures 3.2 and 3.3, respectively. The motive of EGWO is to reduce the overall MOS transistor area while meeting the target specifications of the CMOS DIFFAMP and two-stage miller compensated OPAMP that include A_v, UGB, SR, $ICMR-$, $ICMR+$, P_d and C_L. To mitigate the effects of channel length modulation, the channel length of MOS transistors is taken more than the minimum transistor length i.e., $L_i = 0.7\ \mu m$ for $i = 1\ to\ 8$. The design parameters of DIFFAMP and OPAMP include widths of transistors, capacitance's (C_C and C_L) and I_{bias}. After imposing the appropriate matching properties, transistors M_1, M_3 and M_5 are chosen to be identical to M_2, M_4 and M_8, respectively. The empirical equations used for the design of CMOS DIFFAMP and two-stage OPAMP are explained in A.1 [117].

The initial population size for the EGWO algorithm is considered to be a matrix of size (number of particles (P) × particle vector (Q)), where P = 60 and Q = 7. The particle vector for

the optimal design of CMOS DIFFAMP and two-stage CMOS OPAMP is as follows:

$$X_{opamp} = [A_v, C_L, SR, V_{ICmin}, V_{ICmax}, UGB, P_d] \tag{4.5}$$

$$X_{diffamp} = [A_v, C_L, SR, V_{ICmin}, V_{ICmax}, f_{-3dB}, P_d] \tag{4.6}$$

Here, the cost function is defined as the total MOS area occupied (sum of widths × lengths) by all the transistors which is given as:

$$CF = \sum_{i=1}^{N} \left(S_i \times L_i^2 \right) \tag{4.7}$$

where N is the total number of transistors in a circuit with the desired value of the cost function to be less than 200 μm^2 and 300 μm^2 for CMOS DIFFAMP and two-stage CMOS OPAMP circuits, respectively. Here, the EGWO algorithm is utilized for obtaining the optimal value of the cost function.

4.5 Results and discussion

The optimized design of CMOS DIFFAMP and two-stage OPAMP, shown in Figures 3.2 and 3.3, aims to reduce the overall MOS transistor area with constraints on A_v, C_L, SR, V_{ICmin}, V_{ICmax}, UGB and P_d as given in column 2 of Table 4.4 and Table 4.5, respectively. The aspect ratios of the MOS transistors obtained from the EGWO using MATLAB are employed for the circuit-level implementation of CMOS DIFFAMP and two-stage CMOS OPAMP in Cadence IC616.

Table 4.2: Design parameters for CMOS differential amplifier using EGWO.

Design Parameter	Differential Amplifier			
	WOA	GSA	PSO	EGWO
I_b (μA)	12	20	21	10
$W_{1,2}$ (μm)	7.25	8	10.65	5
$W_{3,4}$ (μm)	1.55	3	2.41	0.7
W_5 (μm)	5.27	3.75	4	3.5
W_6 (μm)	3.12	3.75	4	3.6
L (μm)	0.75	0.75	0.75	0.7

Table 4.2 and Table 4.3 show the optimum design parameters of CMOS DIFFAMP and two-stage OPAMP, respectively, obtained using EGWO and its comparison with other related

Table 4.3: Design parameters for two-stage OPAMP using EGWO.

Design parameters	Simulation-based		Equation-based				
	GSA [118]	AGSA [118]	PSO [119]	WOA [120]	PSO [120]	DE [120]	EGWO
W_1/L_1	4/2	4/2	7.74/0.18	4/2	4/2	4/2	9.4/0.7
W_3/L_3	4/2	4/2	14.4/0.18	5/2	4/2	4/2	2.2/0.7
W_5/L_5	4/2	4/2	1.96/0.18	2/2	4/2	2/2	2.3/0.7
W_6/L_6	21.94/2	7.16/2	98.23/0.18	21.54/2	33.4/2	21.5/2	8/0.7
W_7/L_7	11.36/2	4/2	11.97/0.18	5.38/2	16.7/2	5.38/2	15/0.7
W_8/L_8	4/2	4/2	1.96/0.18	2/2	16.7/2	5.38/2	2.3/0.7
I_{bias} (μA)	45.28	30	34.46	49.0	21	49	17.4
C_C (pF)	4.4	2.2	–	2.45	2.45	2.45	1.65
C_L (pF)	10.02	10	–	7	7	7	7

Units of W_i and L_i are μm, where i = 1 to 8

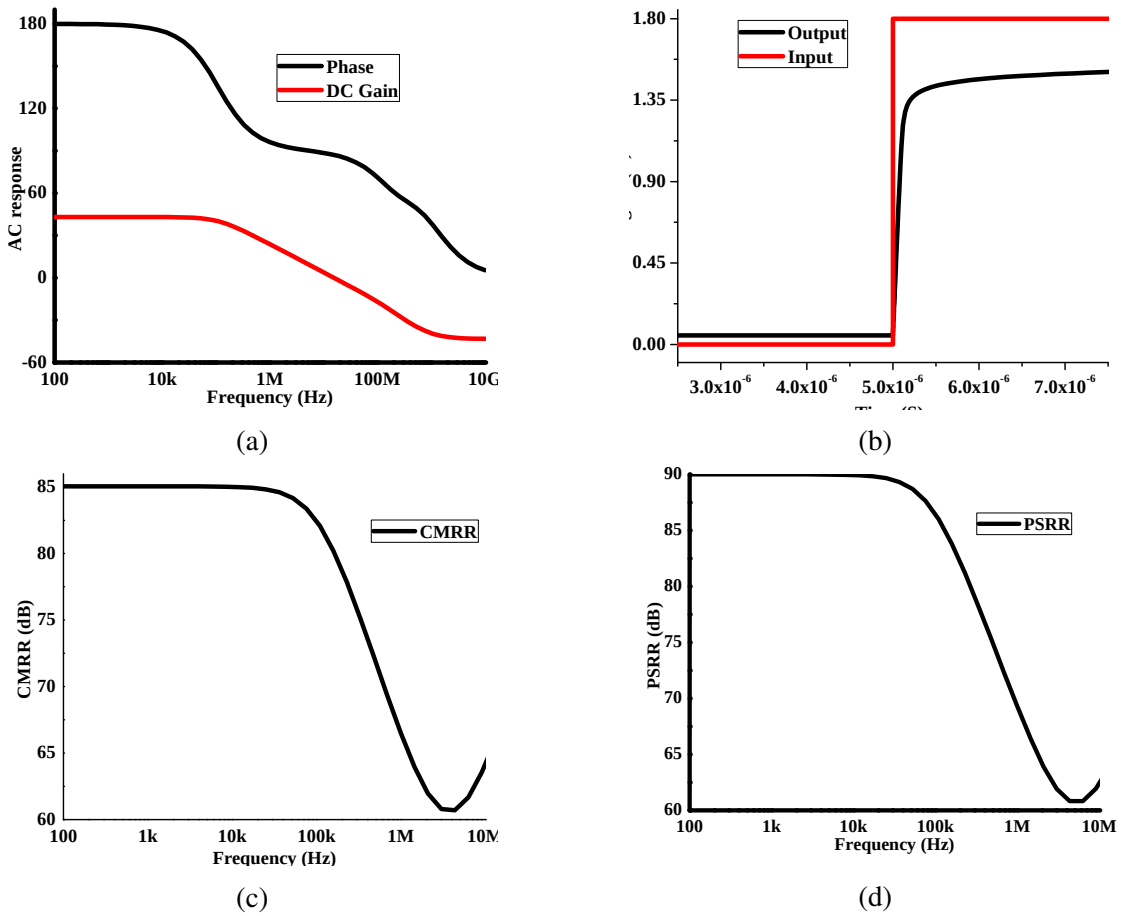
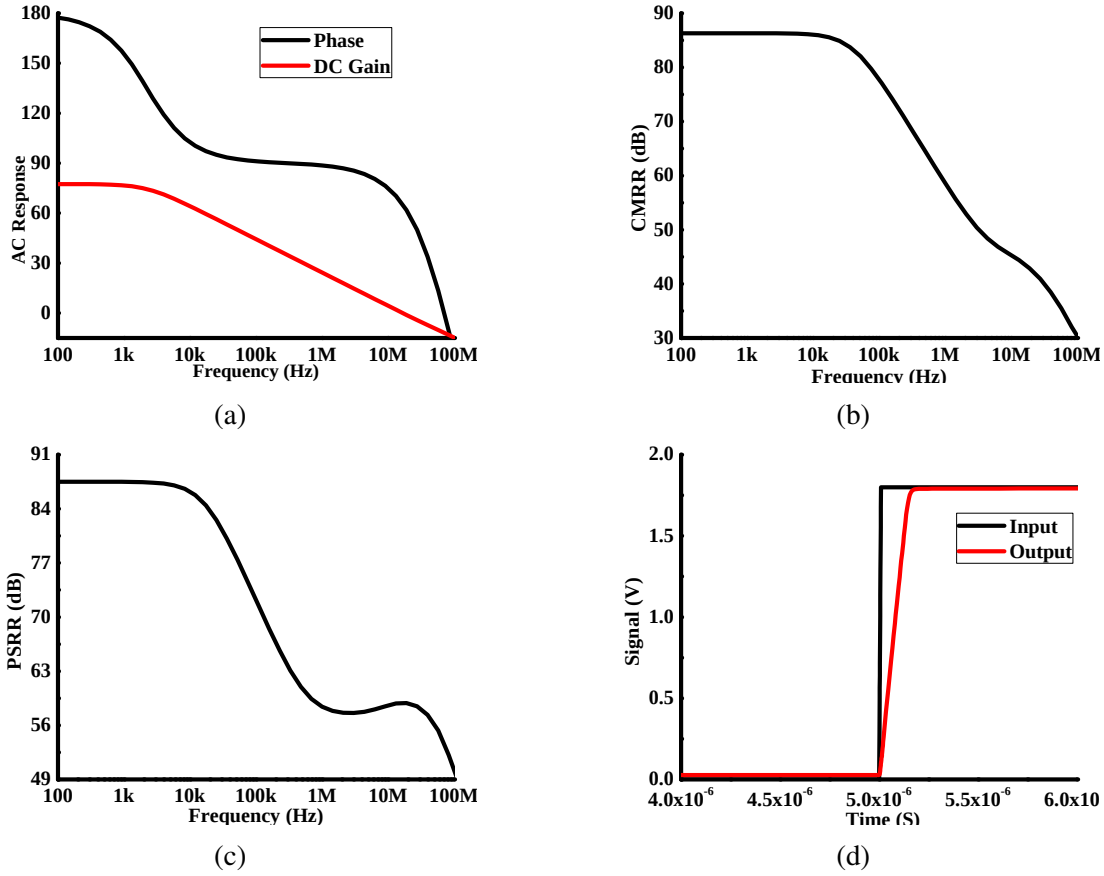


Figure 4.7: Response of CMOS DIFFAMP, using EGWO, to obtain (a) Gain and Phase margin, (b) Slew rate, (c) CMRR and (d) PSRR.

Table 4.4: Performance comparison of differential amplifier.

Design criteria	Perf. spec.	GSA	PSO	WOA	EGWO
Technology (nm)		180 CMOS	180 CMOS	180 CMOS	180 CMOS
SR ($V/\mu s$)	≥ 10	19.59	17.92	14.79	11.57
Pd (μW)	≤ 200	71.62	92.45	70.20	60.75
PM ($^\circ$)	≥ 45	89.44	87.27	89.54	85.40
f_{-3dB} (KHz)	≥ 200	91.2	155.9	154	277.8
Gain (dB)	≥ 40	45.07	44.06	46.06	43.06
CMRR (dB)	≥ 60	82.66	78.5	85.49	85.05
PSRR (dB)	≥ 70	90.09	87.95	92.12	90.12
MOS area (μm^2)	Min.	22.12	25.57	19.83	18.40



(a) (b) (c) (d)

Figure 4.8: Response of two-stage CMOS OPAMP, using EGWO, to obtain (a) Gain and Phase margin, (b) Slew rate, (c) CMRR and (d) PSRR.

Table 4.5: Results obtained using EGWO and its comparison using different algorithms.

Design specifications	Target	PSO [119]	WOA [120]	PSO [120]	DE [120]	GSA [118]	AGSA [118]	EGWO
A_v (dB)	>60	59.19	74.08	61	74.086	60.14	81.13	77.43
GBW (MHz)	>3	3	3	3.85	3	3.136	3.29	16.95
PM $(degrees)$	>45	63.53	–	–	–	47.53	57.91	64.86
SR $(V/\mu s)$	>10	18.35	10	10	20	10.29	12.34	10.05
Pd (mW)	<2.5	0.184	1.137	0.979	2.5	1.053	0.332	0.094
CL (pF)	>7	–	**7**	7	7	10.02	10	7
V_{ICmin} (V)	>0.3	–	-0.01	-0.01	-0.01	-0.86	-1.22	0.4
V_{ICmax} (V)	<1.6	–	**1.1**	1.1	1.1	1.8	1.79	1.2
$CMRR$ (dB)	>60	67.08	–	–	–	81.99	84.6	86.31
$PSRR+$ (dB)	>70	63.84	–	–	–	77.36	97.45	87.5
$PSRR-$ (dB)	>70	99.16	–	–	–	86.82	84.86	78.65
$Area(\mu m^2)$	Objective	28.52	93.86	148.2	93.867	114.6	70.32	35.56
FOM_1	Max.	–	0.43	1.283	0.43	0.69	1.09	6.818
FOM_2	Max.	–	0.197	0.186	0.01	0.26	1.41	3.38
Technology (μm)		0.18	0.18	0.18	0.18	0.35	0.35	0.18

algorithms. The illustration of the optimization trend for optimal design of two-stage OPAMP is shown in the form of convergence plot in Figure 4.6. The convergence plot shows different feasible solutions at different iteration with convergence at an optimal solution (minimum area), which is finally considered for the design of OPAMP. From Tables 4.4 and 4.5, it can be observed that EGWO has the ability to obtain a set of feasible solutions considering designer specific performance requirements. Figure 4.7a to 4.7d show the response of CMOS DIFFAMP to obtain A_v, PM, f_{-3dB}, SR, $CMRR$ and $PSRR$. Similarly, Figures 4.8a to 4.8d show the response of two-stage OPAMP to obtain AC response, SR, $CMRR$ and $PSRR$. As can be observed from the Tables 4.4 and 4.5, the presented EGWO results in better performance as compared to the competing algorithms. For a fair comparison between recent techniques using algorithms such as PSO, WOA, DE, GSA, AGSA, and GWO, with application to circuit sizing that utilize different technologies, Figures Of Merit (FOM) [122], [123] are considered which are given in Equation 3.19 and Equation 3.20.

4.5.1 Statistical Study

Metaheuristic algorithms' capability makes their application in IC sizing more comfortable. But, the random nature of these algorithms stops them from producing constant output for every execution. Therefore, executing these algorithms only once may not be appropriate to comment on their performance. Hence, a statistical study is required to verify the actual perfor-

mance and robustness of the presented method. The results shown in Table 4.4 and Table 4.5 are considered the best solutions obtained by using respective algorithms. The EGWO based optimization process is repeated for 20 times with 20 different initializations of search agents. Table 4.6 shows the actual performance of the proposed algorithm for its application to analog circuit sizing, with each column representing the results obtained by using the best, average and worst solutions from EGWO. Moreover, it is observed that all individual runs lead to feasible solutions which demonstrates the strong exploration ability of EGWO based optimization process. The proposed technique results in low power making it relatively preferable for low power circuit sizing problem.

Table 4.6: Results obtained using EGWO and its comparison using different algorithms.

Design Specifications	EGWO (best)	EGWO (median)	EGWO (worst)
A_v (dB)	77.43	65	78
$GBW(MHz)$	16.95	7.56	7.65
PM(degrees)	64.86	65	60.05
$SR(V/\mu s)$	10.05	10.25	10.02
$Pd(\mu W)$	94.5	167	225
$CL(pF)$	7	7	7
$V_{ICmin}(V)$	0.4	0.4	0.5
$V_{ICmax}(V)$	1.2	1.1	1.2
$CMRR(dB)$	86.31	90	90.51
$PSRR+(dB)$	87.5	104.1	107.08
$PSRR^{\smile}(dB)$	78.65	80.08	80.5
$Area(\mu m^2)$	34.16	77	171.5
FOM_{opamp1}	6.818	2.252	2.142

4.5.2 Corner analysis

The corner analysis is usually performed during the optimization process for validating design efficiency and robustness, considering process corners, temperature and supply voltage variations. Here, the design with best solution obtained is examined for 45 states as a result of cross combinations from five process corners i.e., FF (fast/fast), FS (fast/slow), SS (slow/fast), SF (slow/fast) and TT (typical/typical), supply voltage variations i.e., $V_{dd} \pm 5\%$ and temperature variations i.e., -40°C, 25°C and 90°C. The numerical results of the corner analysis obtained for the design with the best solution obtained using EGWO are provided in Table 4.7. The values shown in Table 4.7 meet the specification requirements of the design for A_v, GBW and PM.

Table 4.7: Results obtained after performing corner analysis on best solution.

Corner	T (°C)	Vdd - 5% (V)			Vdd (V)			Vdd + 5% (V)		
		Av (dB)	GBW (MHz)	PM (°)	Av (dB)	GBW (MHz)	PM (°)	Av (dB)	GBW (MHz)	PM (°)
FF	-40	78.15	11.95	69.2	78.12	13.32	67.54	78.21	13.89	66.95
	25	77.65	13.06	67.56	77.62	15.33	64.46	77.61	14.37	66.3
	90	76.77	14.18	65.92	76.61	15.93	63.65	76.57	15.54	64.57
FS	-40	77.32	10	71.98	78.32	12.66	68.79	78.32	13.04	68.57
	25	77.12	11.87	69.11	77.83	14.2	66.36	7.79	14.37	66.48
	90	76.44	12.86	67.67	96.98	15.33	64.66	76.85	15.54	64.72
SS	-40	76.8	9.6	73.5	78.37	13.67	68.77	77.62	14.94	67.37
	25	77.01	11.44	70.68	77.94	15.33	66.01	76.86	15.54	66.15
	90	76.43	12.86	68.52	77.22	16.13	64.74	75.78	16.16	65.14
SF	-40	78.37	12.49	69.61	77.75	14.38	67.52	77.37	15.54	66.13
	25	78	13.63	67.79	77	16.13	64.66	76.33	15.5	65.87
	90	77.24	14.88	65.91	75.89	16.97	63.32	74.88	16.92	63.7
TT	-40	78.12	11.16	70.99	78.04	14.82	68.04	78.1	13.65	68.28
	25	77.77	12.86	68.25	77.43	16.95	64.85	77.42	15	66.11
	90	77.04	14.84	65.23	76.56	17.05	64.17	76.37	16.32	64.1

4.6 Summary

Over the past years, the automation of analog circuit design and sizing would have been more successful if the optimization algorithms could have incorporated the major trade-offs like accuracy, robustness and run-time simultaneously. This challenge is addressed by introducing an improved algorithm i.e., EGWO, considering it to be a step towards the improved performance of automated sizing tools. Here, the balance between exploration and exploitation is revisited to improve its exploration ability prior to convergence at globally optimal solution. The results obtained after evaluating a set of 23 benchmark functions is compared with algorithms from recent literature revealing its robustness and better performance owing to higher ability of exploration. Despite of the advantages, the EGWO suffers from slightly additional run-time due to inclusion of additional steps in the process of enhancement. The presented circuit sizing methodology is validated considering amplifier circuits, i.e., CMOS DIFFAMP and a two-stage CMOS OPAMP, in 180nm CMOS technology as benchmark circuits, resulting in a reduced area and power consumption, in comparison to foregoing algorithms. To ensure the robustness of EGWO, a statistical study is performed with over 20 independent runs to return a feasible solution in every case. The challenges faced in this method were fine tuning the design parameters to meet the design specifications manually.

Chapter 5

A Hybrid WOA-mGWO Algorithm for Analog Circuit Sizing

5.1 Introduction

The hybridization of WOA and modified GWO (mGWO) algorithm [128] (WOA-mGWO) is proposed and the same is applied for the automated design of analog circuits. The hybrid WOA-mGWO algorithm is tailored in order to improve the exploration ability of the algorithm by combining the abilities of two metaheristic algorithms i.e., WOA and mGWO algorithms. For evaluating the performance of the proposed algorithm, a conventional two-stage OPAMP is considered as a benchmark circuit. The aspect ratios calculated by simulating the algorithm in MATLAB are later used to design the OPAMP in CADENCE analog design environment using 180 nm CMOS standard process.

5.2 Hybrid WOA-mGWO algorithm:

The WOA and mGWO are the metaheuristic algorithms inspired by the social hierarchy and the foraging or hunting behavior of humpback whales and grey wolfs, respectively. In GWO, the exploration and exploitation depends mainly on two parameters i.e., 'a' and 'A'. Half of the iterations are devoted to exploration phase i.e., when $A > 1$ and the other half are assigned to exploitation phase i.e., when $A < 1$. Higher exploration results in lower probability of stagnation at local optimum. The modifications are done keeping in mind the ratio of explo-

ration and exploitation that is to be maintained. The mGWO improves the exploration ability by increasing the non-linearity of parameter 'a' as shown below.

$$a = \begin{cases} 2\left(1 - \frac{t}{T}\right) & ; for\ conventional\ GWO \\ \\ 2\left(1 - \frac{t^2}{T^2}\right) & ; for\ modified\ GWO \end{cases} \tag{5.1}$$

where 't' is the present iteration, 'T' is the maximum number of iterations. The value is chosen to be 2 to dedicate 70 percent of iterations for exploration and the remaining for exploitation as shown in figure.The variation of 'a' over the course of 500 iterations for different values of 'i' is shown in Figure 4.1a. The flowchart shown in Figure 5.1 describes the iterative process of optimization using mGWO. The approach aims to enhance the exploitation of the WOA

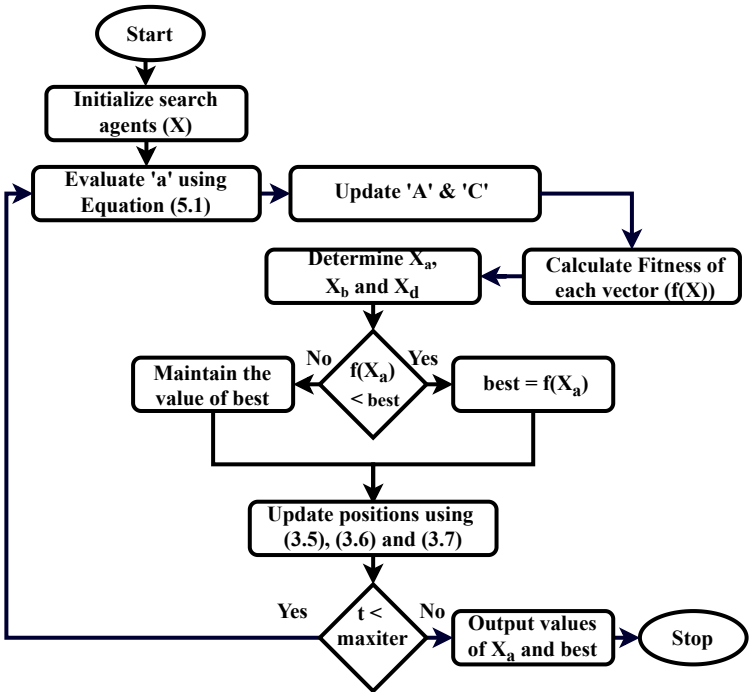


Figure 5.1: Flowchart of mGWO algorithm.

algorithm. To enhance the exploitation, the authors employed mGWO algorithm, with increased exploration ability, in two hybridization models; namely low-level teamwork hybrid (LTH) and low-level relay hybrid (LRH) [129]. In LTH model, mGWO is used as a component in the WOA algorithm. It is used to search the neighborhood of the best search agent so far to insure that its the local optima. In the second model, mGWO is employed in a pipeline mode after the WOA terminates to enhance the best found solution. Upon comparison of the results, the LTH model shows superior performance over the LRH model and hence the LTH model is proposed for the application to the CMOS circuit sizing problem. Here, two algorithms, i.e.,

WOA and mGWO, are merged together to effectively utilize the combined abilities of each algorithm to deal with both local and global search processes adequately. In Hybrid WOA-mGWO, a standard WOA algorithm is used for globally searching the entire search space and to bring most of the solutions towards the favorable area. After the exploration phase, mGWO algorithm is used to locally search for the best (optimal) solution. In Hybrid WOA-mGWO, the WOA algorithm emphasizes on the expansion at the starting point of search to explore the entire search space broadly, while mGWO algorithm focuses on the magnification by allowing the search agents to move towards the best solution as a next stage in the optimization process.

5.2.1 Steps of proposed Hybrid WOA-mGWO algorithm

This section discusses the detailed flow of the proposed hybrid WOA-mGWO algorithm.

Step 1: Initializing the random vector of population size N with dimension D as a starting point of the optimization algorithm.

$$X_{i,j} = lb_j + rand()(ub_j - lb_j) \tag{5.2}$$

Where, $X_{i,j}$ is the position of i^{th} solution and j^{th} dimension; ub_j and lb_j are the upper and lower bounds with j^{th} dimension. Here, 'i' and 'j' varies from 1 to N and 1 to D, respectively; '$rand()$' takes any random number from zero to one.

Step 2: Evaluation of fitness function using each vector $X_{i,j}$ i.e., calculating the values of the function using $X_{i,j}$ for entire population and obtaining the best position.

Step 3: Initialization of algorithm parameters such as 'r_{1j}', 'r_{2j}', 'a_j', 'C_j', 'A_j', 'b' and 'p'. The variables 'r_1', 'r_2' and 'p' take random values between 0 and 1. The coefficient vectors, 'A_j' and 'C_j', are obtained using Equations (5.3) and (5.4), respectively. The parameter 'b' is a constant defining the logarithmic spiral in humpback whales. The components of 'a_j' are calculated as shown in the Equation (5.1) whose value decreases from 2 to 0 with iterations. The variation of a over the course of 500 iterations for different values of 'r' (i.e., 'r' = 1 to 6) is shown in Figure 4.1a that helps in determining the value of 'r' for slightly increasing the number of iterations assigned for exploration. In other words, as the value of 'r' increases, the non linearity increases while improving the overall exploration ability of the algorithm. In order to maintain the balance between exploration and exploration, the value of 'r' is chosen to be 2

in mGWO algorithm.

$$A_j = 2a_j r_{1j} - a_j \tag{5.3}$$

$$C_j = 2r_{2j} \tag{5.4}$$

$$a = 2(1 - \frac{(t)^r}{(Maxiter)^r}) \tag{5.5}$$

where 'r_1' and 'r_2' are the random numbers within the range [0,1], 't' is the current iteration and '$Maxiter$' is the maximum number of iterations.

Step 4: When $p < 0.5$ and $|A| < 1$, the positions of search agents are updated with respect to the three best search agents, i.e., alpha, beta and delta, as shown in Equations (5.7a), (5.7b) and (5.7c) that are denoted by 'X_{ja}', 'X_{jb}' and 'X_{jd}', respectively. In other words, remaining search agents depend on these best search agents to randomly update their positions around the prey.

$$D_{ja} = \left|C_{j1}X_{ja} - X_j\right|, \tag{5.6a}$$

$$D_{jb} = \left|C_{j2}X_{jb} - X_j\right|, \tag{5.6b}$$

$$D_{jd} = \left|C_{j3}X_{jd} - X_j\right| \tag{5.6c}$$

$$X_{1j} = X_{ja} - a_{j1}\left(D_{ja}\right), \tag{5.7a}$$

$$X_{j2} = X_{jb} - a_{j2}\left(D_{jb}\right), \tag{5.7b}$$

$$X_{3j} = X_{jd} - a_{j3}\left(D_{jd}\right) \tag{5.7c}$$

$$X_j(t+1) = \frac{\left(X_{1j} + X_{j2} + X_{3j}\right)}{3} \tag{5.8}$$

where, 'X_j' denotes the present position of the search agent and '$X_j(t+1)$' is the updated position with respect to the three best search agents.

Step 5: When $p < 0.5$ and $|A| \geq 1$, the position update is done with respect to randomly selected search agent for exploration using following equations.

$$D_j = \left|C_j X_{jr} - X_j\right| \tag{5.9}$$

$$X_j(t+1) = X_{jr} - a_j D_j \tag{5.10}$$

Here, '$X_{jr}(t)$' is the position of randomly selected search agent from the population. '$X_j(t)$' and '$X_j(t+1)$' denote the present and updated positions of the search agent, respectively.

Step 6: When $p \geq 0.5$, the positions of the search agents are updated in accordance with the three best search agents as shown below.

$$D_{ja} = \left| C_{j1} X_{ja} - X_j \right|, \tag{5.11a}$$

$$D_{jb} = \left| C_{j2} X_{jb} - X_j \right|, \tag{5.11b}$$

$$D_{jd} = \left| C_{j3} X_{jd} - X_j \right| \tag{5.11c}$$

$$X_{1j} = D_{aj} e^{bl} \cos(2\pi l) + X_{aj}(t), \tag{5.12a}$$

$$X_{j2} = D_{bj} e^{bl} \cos(2\pi l) + X_{bj}(t), \tag{5.12b}$$

$$X_{3j} = D_{dj} e^{bl} \cos(2\pi l) + X_{dj}(t) \tag{5.12c}$$

where, 'l' is a random number within the range [-1,1], 'X_{1j}', 'X_{j2}' and 'X_{3j}' represent the helix shaped movements observed in the search agents (i.e., humpback whales) and the position update is based on the three best search agents as follows:

$$X_j(t+1) = \frac{\left(X_{1j} + X_{j2} + X_{3j} \right)}{3} \tag{5.13}$$

Step 7: If any of the control variables violate the limits, then its value is restricted to lower or upper limit.

Step 8: Checking for termination criteria, i.e., whether the algorithm is run for maximum number of iterations specified.

Step 9: If termination criteria is not satisfied, go to Step 3. Else, print the optimum solution obtained.

The optimization flow of WOA-mGWO algorithm can be better explained in the form of a flowchart as shown in the Figure 5.2.

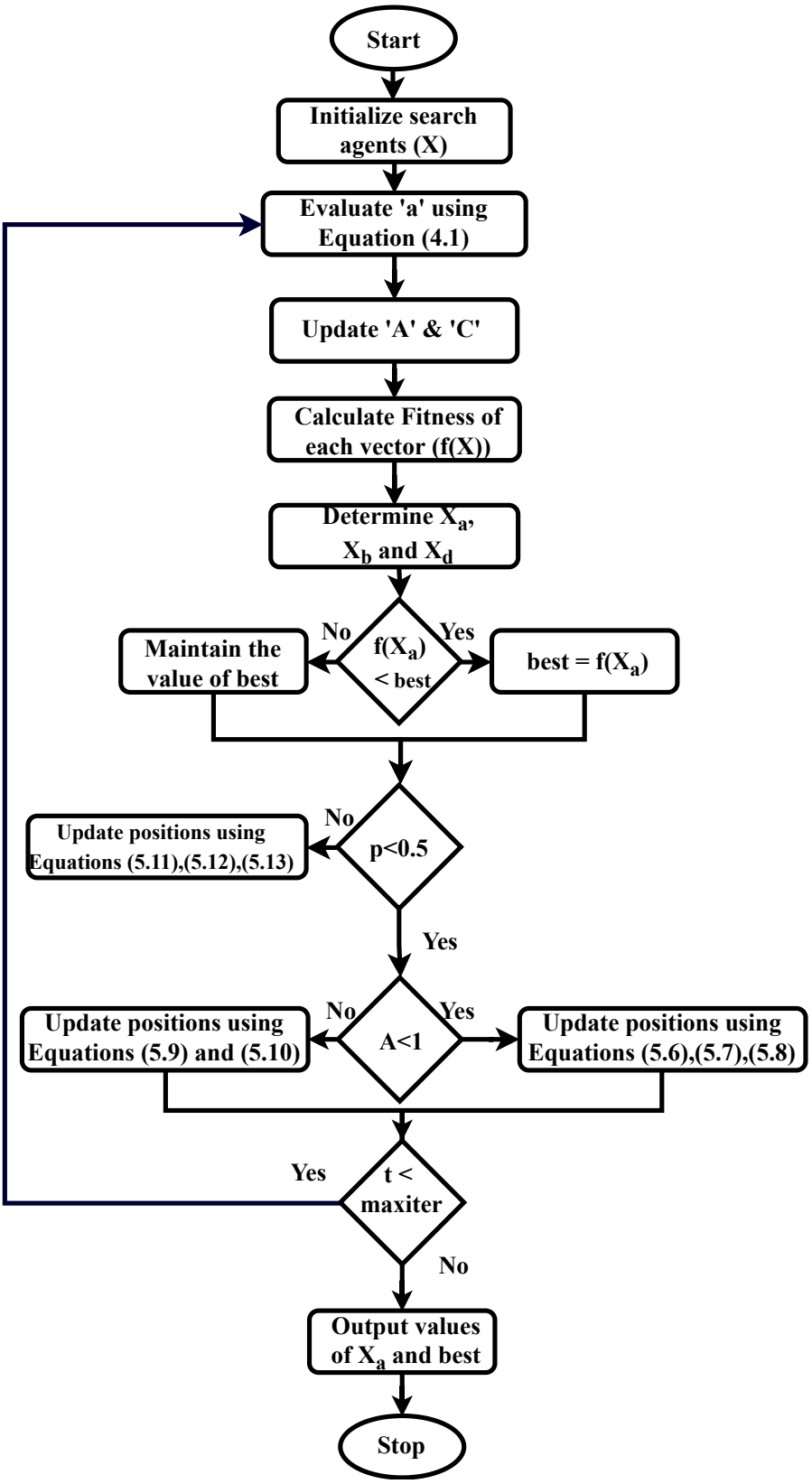


Figure 5.2: Optimization flow for hybrid WOA-mGWO algorithm.

Minimization results of 23 benchmark functions over 20 independent runs for $F1 - F23$.

		SCA	WOA	PSO	GWO	mGWO	GSA PSO	WOA-mGWO
F_1	M	1.54E-19	0	1.80E-37	1.40E-263	1.77E-203	1.53E-19	**0**
	B	7.18E-28	0	2.30E-50	8.94E-269	3.40E-210	1.03E-19	**0**
	W	3.01E-18	0	3.32E-36	1.25E-262	3.32E-202	2.11E-19	**0**
	SD	1.85E-19	0	7.33E-37	0	0	2.76E-20	**0**
F_2	M	9.07E-22	2.03E-322	2.88E-20	9.09E-150	7.04E-119	1.61E-09	**0**
	B	2.08E-26	1.40E-322	1.25E-24	7.65E-153	5.12E-121	1.43E-09	**0**
	W	6.19E-21	2.80E-322	2.78E-19	1.30E-148	3.82E-118	1.90E-09	**0**
	SD	1.10E-19	0	5.88E-18	9.86E-150	8.6E-162	1.27E-10	**0**
F_3	M	64.6287	72.4	0.0116	7.18E-79	1.33E-52	833.33	**1.23E-87**
	B	0.0004	0.5546	0.0028	2.16E-97	3.20E-66	7.93E-19	**5.3E-142**
	W	836.82	353.3114	0.0358	1.44E-77	1.59E-51	6666.66	**1.67E-86**
	SD	68.0306	164.3779	0.0089	8.90E-85	2.64E-88	2059.06	**2.75E-85**
F_4	M	0.3390	15.3559	0.0105	4.20E-66	3.84E-53	8.2791	**9.3E-196**
	B	0.00019	3.93E-19	0.0023	4.81E-69	2.97E-56	1.70E-10	**4.3E-207**
	W	3.1836	80.7672	0.0389	4.83E-65	2.77E-52	82.1799	**1.8E-194**
	SD	0.4404	19.4109	0.0055	6.13E-66	1.51E-77	24.952	**0**
F_5	M	94.8308	24.3639	4.75E+01	25.9161	26.7851	**20.492**	24.3451
	B	26.6304	24.0293	1.53E+01	24.3029	26.1738	**4.1395**	23.8722
	W	1378.96	**24.9405**	8.74E+01	27.0720	28.5489	73.777	26.9609
	SD	0.6309	**0.2106**	2.78E+01	0.6472	0.7516	13.093	0.2660
F_6	M	3.5692	6.06E-06	**1.96E-32**	0.1628	0.6141	505.0125	7.43E-07
	B	2.6355	2.90E-06	**0**	4.28E-07	0.2502	1.18E-19	2.88E-07
	W	4.0905	1.27E-05	**6.78E-32**	0.5030	1.4991	10100.25	1.18E-06
	SD	0.3552	4.07503	**2.60E-32**	0.228	0.4033	2258.48	5.59E-02
F_7	M	0.0035	1.49E-04	8.77E-03	0.0001	3.23E-04	0.00906	**6.71E-05**
	B	0.0002	6.14E-06	4.84E-03	7.83E-06	1.07E-04	0.0033	**3.08E-06**
	W	0.0294	5.77E-04	1.46E-02	0.0004	7.57E-04	0.0184	**1.45E-04**
	SD	0.004	2.6E-04	3.17E-03	4.61E-05	2.56E-05	0.0039	**4.21E-05**
F_8	M	-4363.6	-11958.18	-6855.10	-6282.65	-5705.05	-8401.25	**-12011.81**
	B	-5061.8	-12569.48	-8798.60	-7639.64	-6501.72	-9441.42	**-12569.48**
	W	-4034.8	-8569.20	-5462.02	-5288.60	-4901.58	-7195.47	**-8739.81**
	SD	**235.92**	306.89	620.75	425.84	1446.42	723.46	820.85
F_9	M	1.991	0	23.5307	0	0	105.11	**0**
	B	0	0	16.9142	0	0	47.757	**0**
	W	38.86	0	38.8033	0	0	142.27	**0**
	SD	0.925	0	4.9232	0	0	24.787	**0**
F_{10}	M	11.375	3.73E-15	1.14E-14	7.82E-15	7.82E-15	1.8604	**3.20E-15**
	B	7.99E-15	8.88E-16	7.99E-15	4.44E-15	4.44E-15	2.32E-10	**8.88E-16**
	W	20.1743	4.44E-15	1.51E-14	7.99E-15	7.99E-15	19.0521	**4.44E-15**
	SD	9.187	2.15E-15	3.51E-15	1.81E-15	1.79E-15	5.7282	**1.67E-15**
F_{11}	M	6.43E-08	0	0.0116	0	1.29E-03	0.00959	**0**
	B	0	0	0	0	0	0	**0**
	W	1.06E-06	0	0.0320	0	1.3E-03	0.04658	**0**
	SD	0.018	4.1E-03	0.0138	0	0	0.01186	**0**

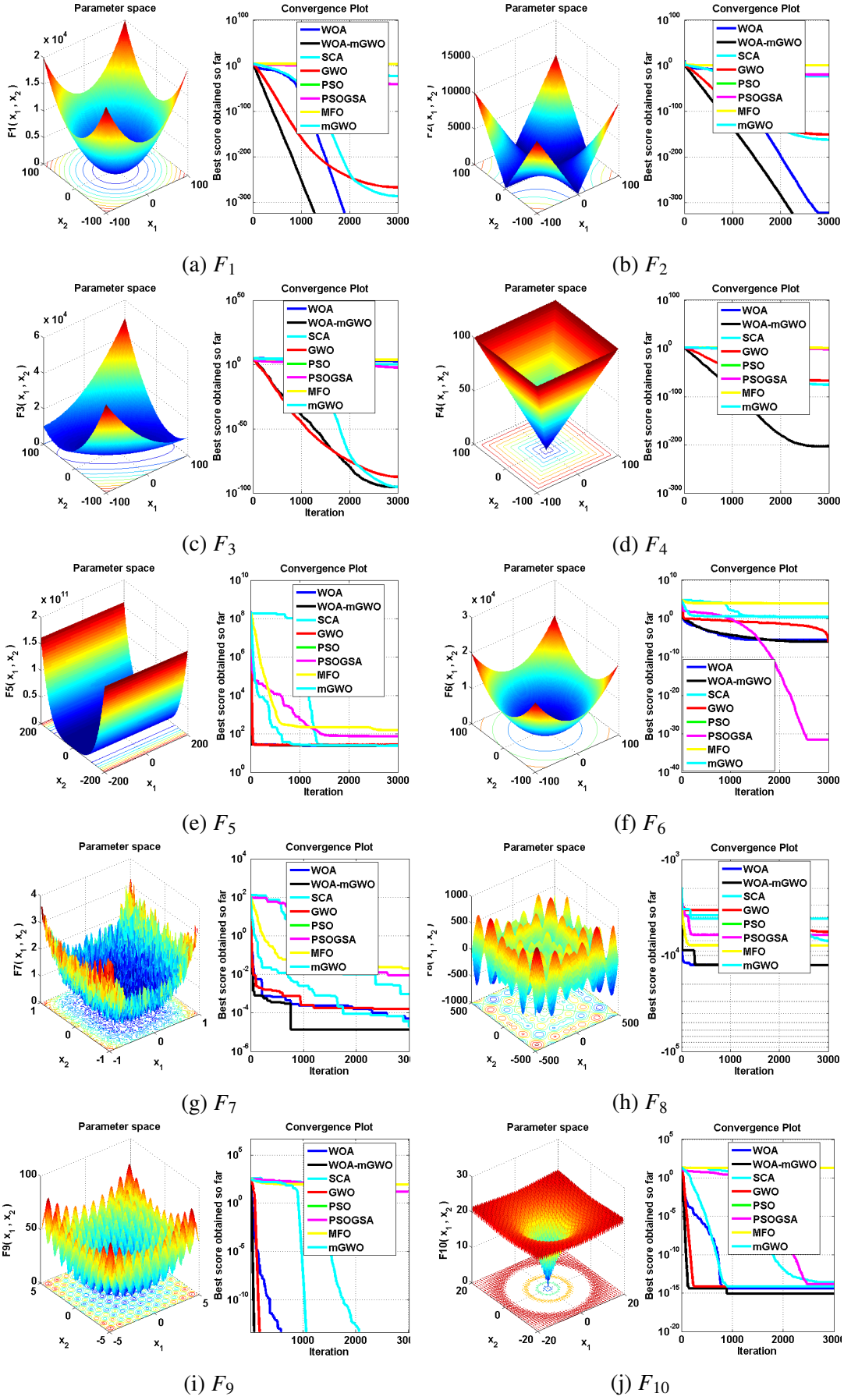
F_{12}	M	0.3185	9.56E-07	**1.68E-32**	0.0132	0.0502	0.8102	0.0016
	B	0.200	5.19E-07	**1.58E-32**	0.0011	0.0196	1.29E-21	7.40E-08
	W	0.4102	1.86E-06	**2.12E-32**	0.0337	0.1434	1.9775	0.0065
	SD	0.0468	4.14E-07	**6.20E-34**	0.0107	0.05115	0.65774	0.0023
F_{13}	M	2.0002	5.6E-04	**5.49E-04**	0.1557	0.5299	0.00054	0.0235
	B	1.798	3.35E-06	**1.47E-32**	7.60E-07	0.1983	2.02E-20	2.13E-06
	W	2.1918	0.0109	**1.10E-02**	0.4757	0.8295	0.01098	0.2479
	SD	0.227	3.19E-05	**1.16E-31**	0.1208	0.1473	0.00245	0.0538
F_{14}	M	0.998	0.998	1.2962	2.1806	4.2252	1.0477	**0.998**
	B	0.998	0.998	0.9980	0.9980	0.9980	0.998	**0.998**
	W	0.998	0.998	1.9920	10.763	12.6705	1.9920	**0.998**
	SD	2.2241	0.3059	5.23E-07	2.69E-14	2.2353	0.22227	**2.24E-14**
F_{15}	M	0.0006	5.87E-04	4.87E-04	0.0033	0.0044	0.00436	**3.19E-04**
	B	0.0003	3.1E-04	3.07E-04	0.0003	0.0003	0.00030	**3.07E-04**
	W	0.0012	1.22E-03	1.05E-3	0.0203	0.0203	0.02036	**5.16E-04**
	SD	0.00035	2.9E-04	3.35E-04	0.0061	9.21E-10	0.008209	**2.81E-04**
F_{16}	M	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	**-1.0316**
	B	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	**-1.0316**
	W	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	**-1.0316**
	SD	4.15E-06	1.28E-14	2.28E-16	2.73E-10	2.80E-06	2.28E-16	**1.28E-16**
F_{17}	M	0.3980	0.3978	0.3978	0.3978	0.3978	0.3978	**0.3978**
	B	0.3978	0.3978	0.3978	0.3978	0.3978	0.3978	**0.3978**
	W	0.3989	0.3978	0.3978	0.3978	0.3978	0.3978	**0.3978**
	SD	9.82E-05	9.33E-10	**0**	1.34E-08	2.53E-05	1.54E-07	1.73E-10
F_{18}	M	3	3	**3**	3	3	3	3
	B	3	3	**3**	3	3	3	3
	W	3	3	**3**	3	3	3	3
	SD	2.50E-07	2.14E-09	**5.09E-16**	7.32E-08	3.29E-07	6.60E-16	1.39E-06
F_{19}	M	-3.8556	-3.8627	**-3.8627**	-3.8623	-3.8619	-3.8627	-3.8627
	B	-3.8626	-3.8627	**-3.8627**	-3.8627	-3.8627	-3.8627	-3.8627
	W	-3.8546	-3.8627	**-3.8627**	-3.8549	-3.8559	-3.8627	-3.8626
	SD	0.00236	1.76E-03	**2.28E-15**	4.13E-07	0.0023	2.24E-15	5.58E-05
F_{20}	M	-3.0476	-3.2585	-3.2744	-3.2585	-3.2604	-3.2506	**-3.2979**
	B	-3.1881	-3.3219	-3.3219	-3.3219	-3.3219	-3.3219	**-3.3219**
	W	-2.6277	-3.0838	-3.2031	-3.1390	-3.0866	-3.2031	**-3.2006**
	SD	0.1274	0.0629	6.0E-02	0.0662	0.0598	0.05975	**5.80E-02**
F_{21}	M	-5.2182	**-10.1531**	-8.8855	-9.6479	-9.3927	-6.00519	-9.8983
	B	-10.0163	**-10.1531**	-10.1531	-10.1531	-10.1532	-10.1531	-10.1532
	W	-0.8805	**-10.1531**	-5.0551	-5.1007	-5.05519	-2.6304	-5.05519
	SD	2.5812	2.58E-07	2.0781	1.8509	1.5551	3.2810	**1.63E-07**
F_{22}	M	-5.5013	-10.4029	-10.1392	-10.4029	-10.1389	-6.7722	**-10.4029**
	B	-8.9717	-10.4029	-10.4029	-10.4029	-10.4029	-10.402	**-10.4029**
	W	-0.9097	-10.4029	-5.1288	-10.4028	-5.12861	-1.83759	**-10.4029**
	SD	1.75E-06	1.75E-06	1.623339	1.1885	2.5810	3.7993	**1.58E-06**
F_{23}	M	-6.2499	**-10.5364**	-10.5360	-10.5363	-10.5361	-5.9184	-9.9956
	B	-9.4500	-10.5364	**-10.5364**	-10.5364	-10.5363	-10.5364	-10.5364
	W	-4.9950	**-10.5364**	-10.5298	-10.5363	-10.5356	-1.8594	-5.1284
	SD	1.6314	3.76E-06	1.1987	1.22E-05	2.200	3.6210	**4.28E-07**

Table 5.2: P-values calculated for the Wilcoxon rank-sum test

Function	GWO	PSO	SCA	WOA	MFO	PSO GSA	WOA-mGWO
F_1	8.01E-09	8.01E-09	8.01E-09	NA	8.01E-09	8.01E-09	NA
F_2	8.01E-09	8.01E-09	8.01E-09	7.90E-09	7.42E-09	8.01E-09	NA
F_3	4.70E-03	6.80E-08	6.80E-08	6.80E-08	6.80E-08	6.80E-08	NA
F_4	6.80E-08	6.80E-08	6.80E-08	6.80E-08	6.80E-08	6.80E-08	NA
F_5	9.13E-07	2.85E-01	6.80E-08	2.39E-01	0.3369	9.75E-06	NA
F_6	6.60E-08	NA	6.60E-08	6.60E-08	6.70E-08	6.56E-08	6.60E-08
F_7	3.24E-01	6.80E-08	6.80E-08	5.65E-02	6.80E-08	6.80E-08	NA
F_8	6.80E-08	6.80E-08	6.80E-08	2.00E-03	6.80E-08	6.80E-08	NA
F_9	NA	7.99E-09	9.60E-03	NA	8.01E-09	8.01E-09	NA
F_{10}	4.28E-09	2.37E-08	4.21E-08	8.90E-01	3.96E-08	2.86E-08	NA
F_{11}	NA	3.27E-06	4.50E-03	NA	3.30E-06	4.14E-04	NA
F_{12}	6.52E-08	NA	6.52E-08	6.52E-08	6.65E-08	6.57E-08	6.52E-08
F_{13}	6.64E-08	NA	6.64E-08	6.64E-08	5.90E-08	1.18E-06	6.64E-08
F_{14}	2.69E-04	8.40E-05	1.38E-07	6.45E-01	7.83E-09	5.52E-04	NA
F_{15}	1.60E-05	7.70E-03	4.68E-05	4.39E-02	0.1198	0.0273	NA
F_{16}	6.64E-08	7.77E-09	6.64E-08	2.03E-01	7.77E-09	7.42E-09	NA
F_{17}	1.43E-07	7.99E-09	6.79E-08	7.64E-02	1.05E-07	2.50E-06	NA
F_{18}	1.51E-08	NA	1.51E-08	1.51E-08	1.42E-04	4.02E-01	3.33E-09
F_{19}	8.01E-09	NA	8.01E-09	8.01E-09	NA	NA	8.01E-09
F_{20}	1.40E-01	1.20E-01	6.80E-08	0.0239	0.4312	3.17E-02	NA
F_{21}	6.80E-08	2.78E-01	6.80E-08	NA	0.1014	0.031	2.29E-01
F_{22}	6.80E-08	1.06E-04	6.80E-08	0.0016	6.80E-05	0.1707	NA
F_{23}	6.80E-08	1.06E-04	6.80E-08	0.0016	2.86E-08	1.47E-01	NA

5.3 Performance evaluation of hybrid WOA-mGWO algorithm

The performance of the hybrid WOA-mGWO is compared with other algorithms with respect to 23 classical and popular benchmark functions [125], as shown in Appendix A.2, to analyze the efficiency of the proposed algorithm. The benchmark functions can be categorized into three groups: Unimodal, Multimodal high-dimensional, and fixed dimension (low-dimensional) multimodal benchmark functions. The Hybrid WOA-mGWO algorithm and other algorithms, namely, SCA, WOA, PSO, GWO, mGWO and a hybrid of Gravitational Search Algorithm (GSA) and PSO algorithms (GSAPSO) [130], are simulated twenty times on each of the twenty three benchmark functions for comparison. The results comprising several statistical parameters, such as mean, standard deviation, median and worst of the best-so-far solution are reported. The results of the algorithms on all test functions are presented in Table 5.1 for functions $F_1 - F_{23}$. To verify the efficiency of the proposed algorithm, the same maximum number of



(a) F_1 (b) F_2

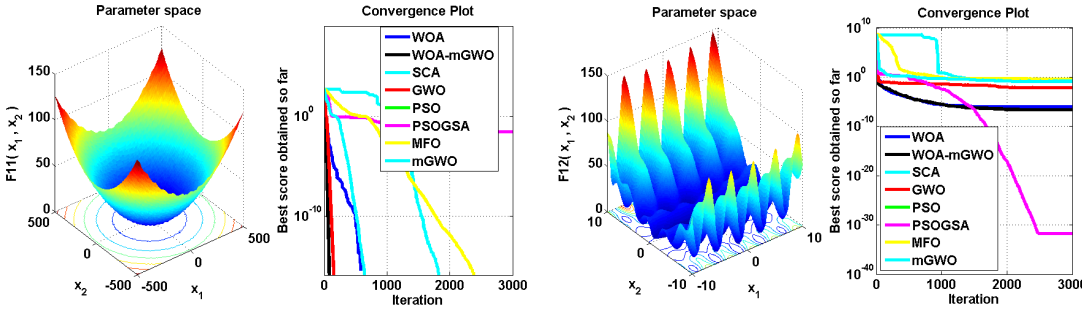
(c) F_3 (d) F_4

(e) F_5 (f) F_6

(g) F_7 (h) F_8

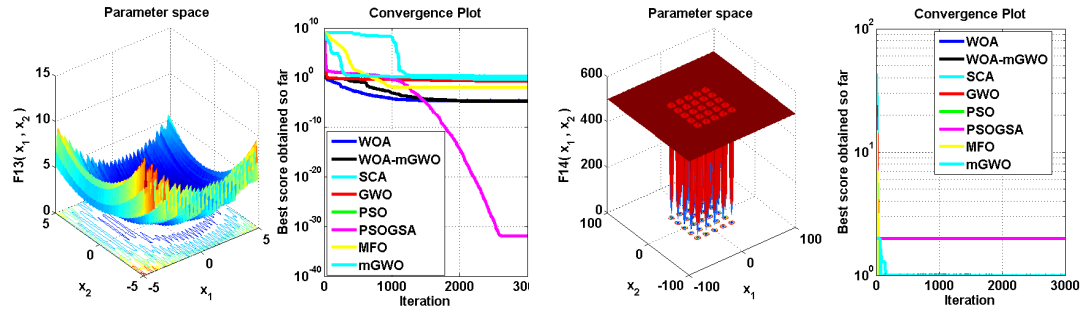
(i) F_9 (j) F_{10}

Figure 5.3: Convergence plots of benchmark functions (F_1 - F_{10}) for 3000 iterations.



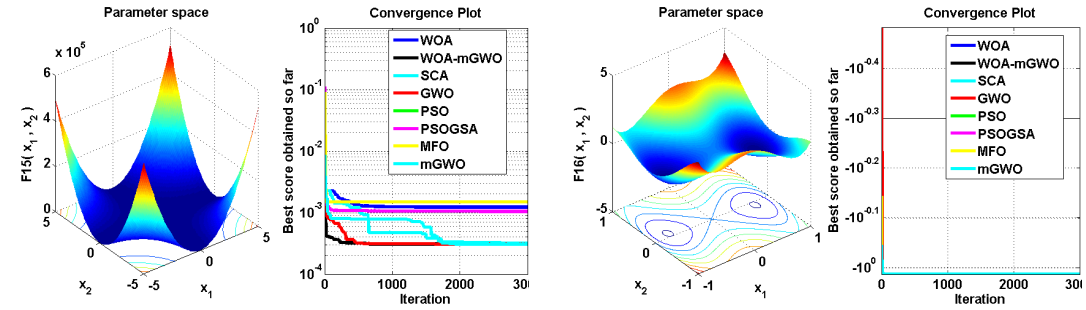
(a) F_{11}

(b) F_{12}



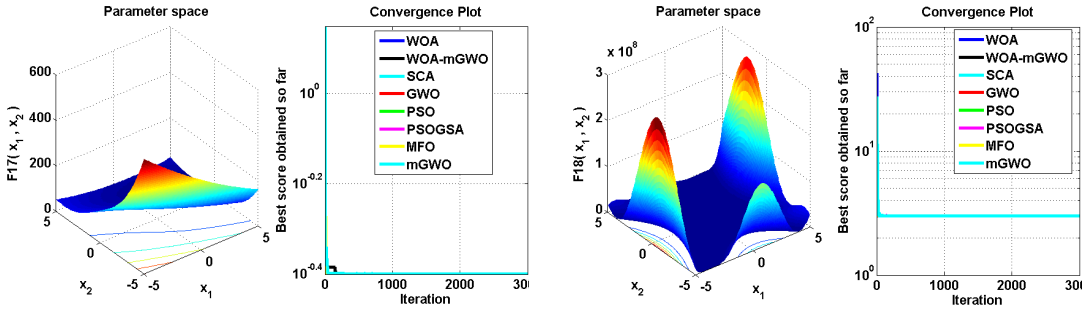
(c) F_{13}

(d) F_{14}



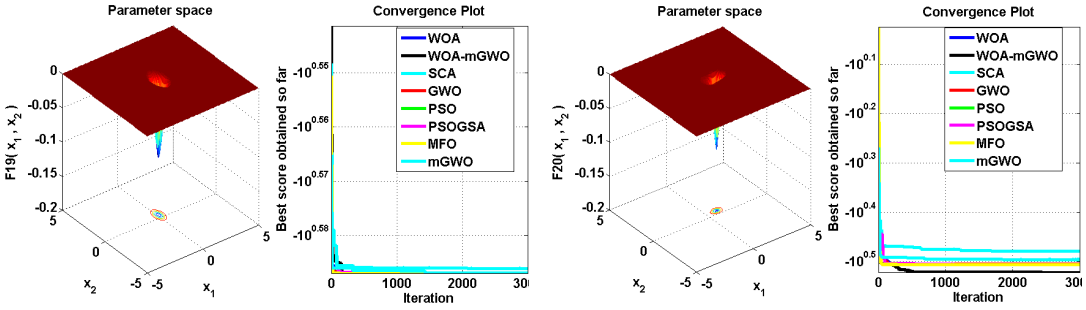
(e) F_{15}

(f) F_{16}



(g) F_{17}

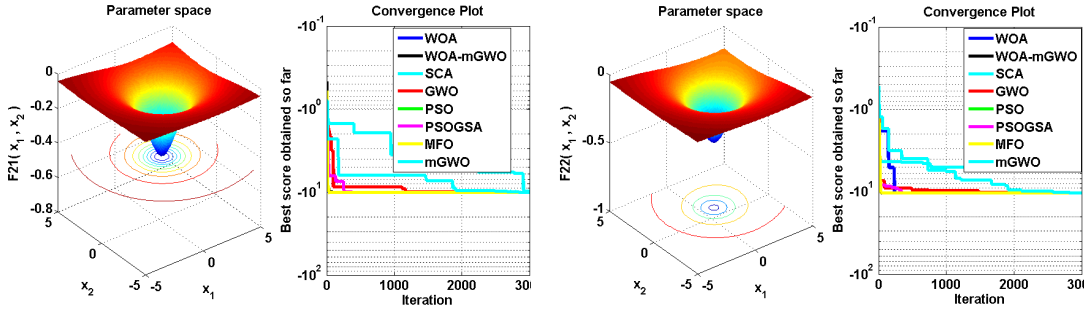
(h) F_{18}



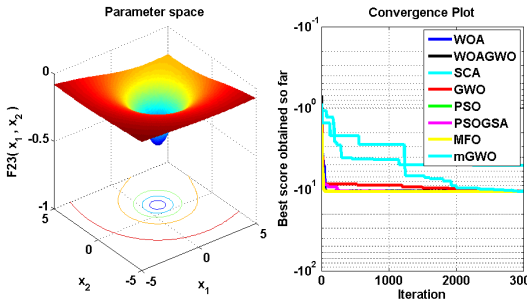
(i) F_{19}

(j) F_{20}

Figure 5.4: Convergence plots of benchmark functions (F_{11} - F_{20}) for 3000 iterations.

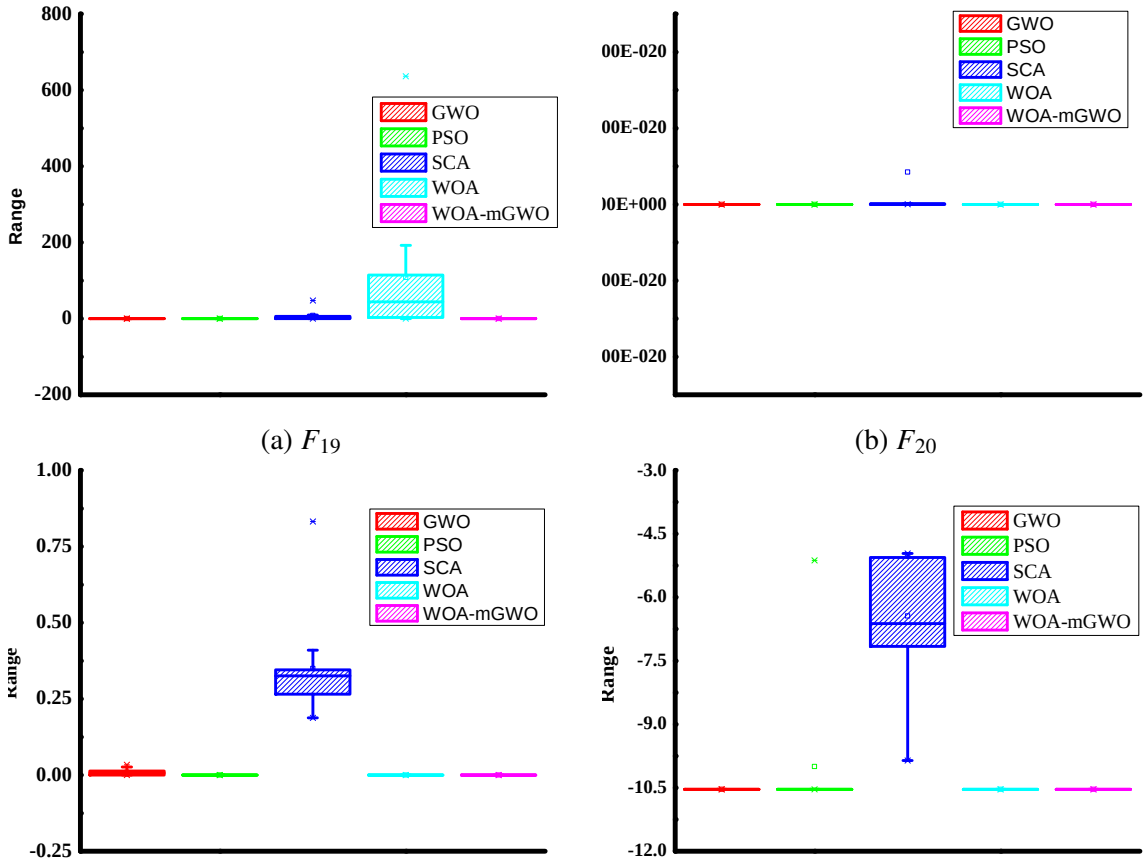


(a) F_{21} (b) F_{22}



(c) F_{23}

Figure 5.5: Convergence plots of benchmark functions (F_{21} - F_{23}) for 3000 iterations.



(a) F_{19} (b) F_{20}

(c) F_{21} (d) F_{22}

Figure 5.6: Box plots for comparing GWO, PSO, SCA, WOA and WOA-mGWO for some of the benchmark functions (F_2, F_7, F_{12}, F_{22}).

iterations and population sizes are taken for the proposed and the existing algorithms to verify the efficiency of the proposed algorithm. The outcomes are averaged over 20 independent runs, and the best outcomes are shown in bold type in the Table 5.1.

The unimodal functions have no local solution and there is only one global solution. Consequently, they are used to examine heuristic optimization algorithms in terms of convergence rate. Functions $F_1 - F_7$ are unimodal functions, and the results for these unimodal functions are demonstrated in Table 5.1. As can be seen in the table, the Hybrid WOA-mGWO outperforms the other algorithms on five out of seven for the unimodal benchmark functions. Therefore, this is evidence that the proposed algorithm has high performance in finding the global solution of unimodal benchmark functions. For multimodal functions, $F_8 - F_{13}$, with many local minima, the final results are more important because of this function can reflect the algorithms ability to escape from poor local optima and obtain the near-global optimum. The functions $F_8 - F_{13}$ where the number of local minima increases exponentially as long as the dimension of the function increases are investigated similarly. The results of mean, standard deviation, median, and the worst value demonstrate that the Hybrid WOA-mGWO performs better than the other algorithms on four out of six multimodal high-dimensional benchmark functions. For $F_{14} - F_{23}$ with only a few local minima, the dimension of the function is also small. The major difference compared with functions $F_8 - F_{13}$ is that functions $F_{14} - F_{23}$ appear to be simpler than $F_8 - F_{13}$ due to their low dimensionalities and a smaller number of local minima. According to the results shown in the table, the Hybrid WOA-mGWO performs better than the other algorithms on seven out of ten of the multimodal low-dimensional benchmark functions. Box plots for the result of 20 independent runs which are computed by using the Hybrid WOA-mGWO, SCA, WOA, PSO and GWO, for functions F_1, F_3, F_{12} and F_{22} are shown in Figure 5.6 demonstrate the range or deviation across the mean value.

Another metric employed to confirm the convergence of the Hybrid WOA-mGWO algorithm is convergence rate. Figure 5.3, Figure 5.4 and Figure 5.5 show the convergence rates of the Hybrid WOA-mGWO, SCA, WOA, PSO, GWO, mGWO and GSAPSO algorithms for functions $F_1 - F_{23}$. The fitness of the best solution in each iteration is saved and drawn as the convergence curves in Figure 5.3, Figure 5.4 and Figure 5.5. The descending trend is quite evident in the convergence curves of Hybrid WOA-mGWO on many of the test functions investigated. This strongly proves the ability of the Hybrid WOA-mGWO algorithm in obtaining a better approximation of the global optimum over the course of iterations. Therefore, it can be stated that the Hybrid WOA-mGWO can balance exploration and exploitation properly and hence is used for solving analog IC sizing problem. The optimum value that can be obtained for all 23 functions is displayed in the last column of table in Appendix A.2. Hence, the ef-

feciency of the algorithm is also decided by the convergence curves of respective algorithm. It can be observed from the convergence plots shown in Figure 5.3 and Figure 5.5 that the proposed WOA-mGWO algorithm converges much faster than the other algorithms proving its superiority.

Comparison of the algorithms based on best, mean, worst and standard deviation values over 20 independent runs cannot be compared for each of the individual runs. Hence, there is always a possibility that the predominance may have occurred accidentally even with low probability in 20 runs. Therefore, further investigation is done using the Wilcoxon statistical test [125] at 5 % significance level and the superiority is decided by comparing the p-values for each run, as reported in Table 5.2. As the best algorithm cannot be compared with itself, it is represented with Not Applicable (NA), for each function. However, the hybrid WOA-mGWO results in p-values much less than 0.05 for most of the benchmark functions demonstrating the statistical significance of the algorithm, except for function F_{21}. Thus, the overall performance of the hybrid WOA-mGWO algorithm proves its dominance over other competing algorithms in the literature.

5.4 Formulation of cost function

The circuit-level implementation and the optimization of the geometrical ratios of MOS transistors for analog circuits employing the hybrid WOA-mGWO algorithm is discussed in this section. Here, the constraints and technology parameters are initialized to obtain the design parameters i.e., aspect ratios of the transistors, bias current (I_{bias}) and capacitor values. The process starts with initialization of parameters randomly within the range specified by the designer followed by focusing on the optimization of the cost function. The algorithm results in new values of the design parameters for optimal value of cost function when the termination condition is satisfied.

The circuit selected as a case study for optimization is a two-stage conventional CMOS OPAMP with miller-compensated topology as shown in Figure 3.3. The motive of hybrid WOA-mGWO algorithm is to reduce the overall MOS transistor area while meeting the target specifications of the OPAMP that include A_v, UGB, SR, $ICMR-$ and $ICMR+$, P_d and C_L. To mitigate the effects of channel length modulation, the channel length of MOS transistors is taken more than the minimum transistor length i.e., $L_i = 1\ \mu m$ for $i = 1\ to\ 8$. The design parameters of OPAMP include widths of transistors (W_i for $i = 1\ to\ 8$), capacitance's (C_C and C_L) and I_{bias}.

After imposing the appropriate matching properties, transistors M_1, M_3 and M_5 are chosen to be identical to M_2, M_4 and M_8, respectively. The empirical equations used for the design of OPAMP are explained in Appendix A.1 [117].

The initial population size for the hybrid WOA-mGWO algorithm is considered to be a matrix of size (number of particles (P) × particle vector (Q)), where P = 60 and Q = 7. The particle vector for the optimal design of two-stage CMOS OPAMP is as follows:

$$X_{opamp} = [A_v, C_L, SR, V_{ICmin}, V_{ICmax}, UGB, P_d] \tag{5.14}$$

Here, the cost functions, CF_1 and CF_2 are defined as the total MOS area occupied (sum of widths × lengths) by all the transistors and the overall power consumption of the circuit, which are combined using apriori weighted sum approach by setting equal weights to both the cost functions. Besides, simple normalization is used for normalizing the values of CF1 and CF2 considering the differences in the units of individual cost functions. The cost functions, CF_1 and CF_2, are given as:

$$CF_1 = \sum_{i=1}^{N} \left(S_i \times L_i^2\right) \tag{5.15}$$

$$CF_2 = V_{DD}(I_{d5} + I_{d7}) \tag{5.16}$$

$$CF = CF_1 + CF_2 \tag{5.17}$$

where S_i is the aspect ratio of the i^{th} transistor, L_i is the length of the i^{th} transistor and N is the total number of transistors in a circuit with the desired value of the cost function to be less than 300 μm^2, for the given circuit. The hybrid WOA-mGWO algorithm is utilized for obtaining the optimal value of the cost function.

5.5 Results and discussion

The optimized design of two-stage OPAMP aims to reduce the overall MOS transistor area with constraints on A_v, C_L, SR, V_{ICmin}, V_{ICmax}, UGB and P_d as given in column 2 of Table 5.4. The constraint handling technique used for optimizing the constrained area optimization problem is static penalty function which is one of the most commonly used constraint handling techniques for circuit sizing tools [131]. The aspect ratios of MOS transistors obtained from the hybrid WOA-mGWO using MATLAB are employed for the circuit-level implementation of OPAMP in Cadence IC616.

Table 5.3: Design parameters obtained using hybrid WOA-mGWO and its comparison using different algorithms.

Design parameters	Simulation-based		Equation-based				
	GSA [118]	AGSA [118]	PSO [119]	WOA [120]	PSO [120]	DE [120]	WOA-mGWO
W_1/L_1	4/2	4/2	7.74/0.18	4/2	4/2	4/2	2.06/1
W_3/L_3	4/2	4/2	14.4/0.18	5/2	4/2	4/2	2.06/1
W_5/L_5	4/2	4/2	1.96/0.18	2/2	4/2	2/2	2.06/1
W_6/L_6	21.94/2	7.16/2	98.23/0.18	21.54/2	33.4/2	21.5/2	6.15/1
W_7/L_7	11.36/2	4/2	11.97/0.18	5.38/2	16.7/2	5.38/2	3/1
W_8/L_8	4/2	4/2	1.96/0.18	2/2	16.7/2	5.38/2	2.06/1
I_{bias} (μA)	45.28	30	34.46	49	21	49	20
C_C (pF)	4.4	2.2	–	2.45	2.45	2.45	1.05

Units of W_i and L_i are μm, where i = 1 to 8

Table 5.3 shows the optimum design parameters of a two-stage OPAMP obtained using hybrid WOA-mGWO and other related algorithms. The illustration of the optimization trend for optimal design of two-stage OPAMP is shown in the form of convergence plot in Figure 5.7. The convergence plot shows different feasible solutions at different iteration with convergence at an optimal solution (minimum area), which is finally considered for the design of OPAMP. From Table 5.4, it can be observed that hybrid WOA-mGWO has the ability to obtain a set of feasible solutions considering designer specific performance requirements. Figures 5.8a to 5.8d show the response of two-stage OPAMP to obtain A_v, PM, UGB, SR, $CMRR$ and $PSRR$. The

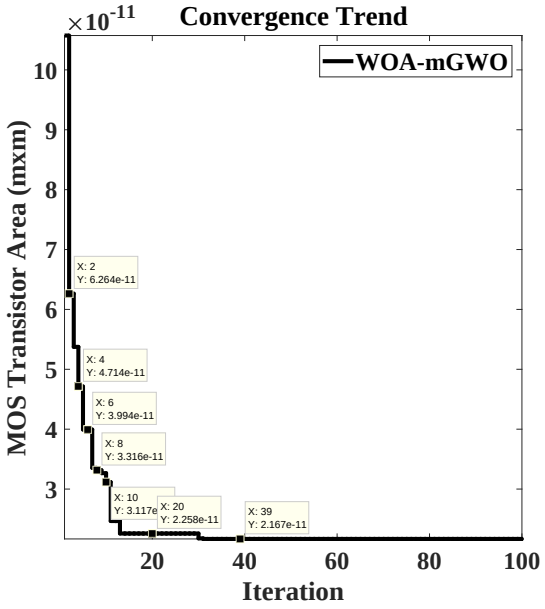


Figure 5.7: Convergence trend for optimal CMOS transistor area using WOA-mGWO.

proposed technique results in low power making it relatively preferable for low power circuit sizing problem. For a fair comparison between recent techniques using algorithms such as, PSO, WOA, DE, GSA, AGSA, and GWO, with application to circuit sizing that utilize different technologies, Figures of Merit(FOMs) are considered which are given in Equation 3.19 and Equation 3.20. As can be observed from the Table 5.4, the presented hybrid WOA-mGWO results in better performance in circuit sizing, aiming at reduction in overall MOS transistor area, when compared to the aforementioned algorithms.

Table 5.4: Results obtained using hybrid WOA-mGWO and its comparison using different algorithms.

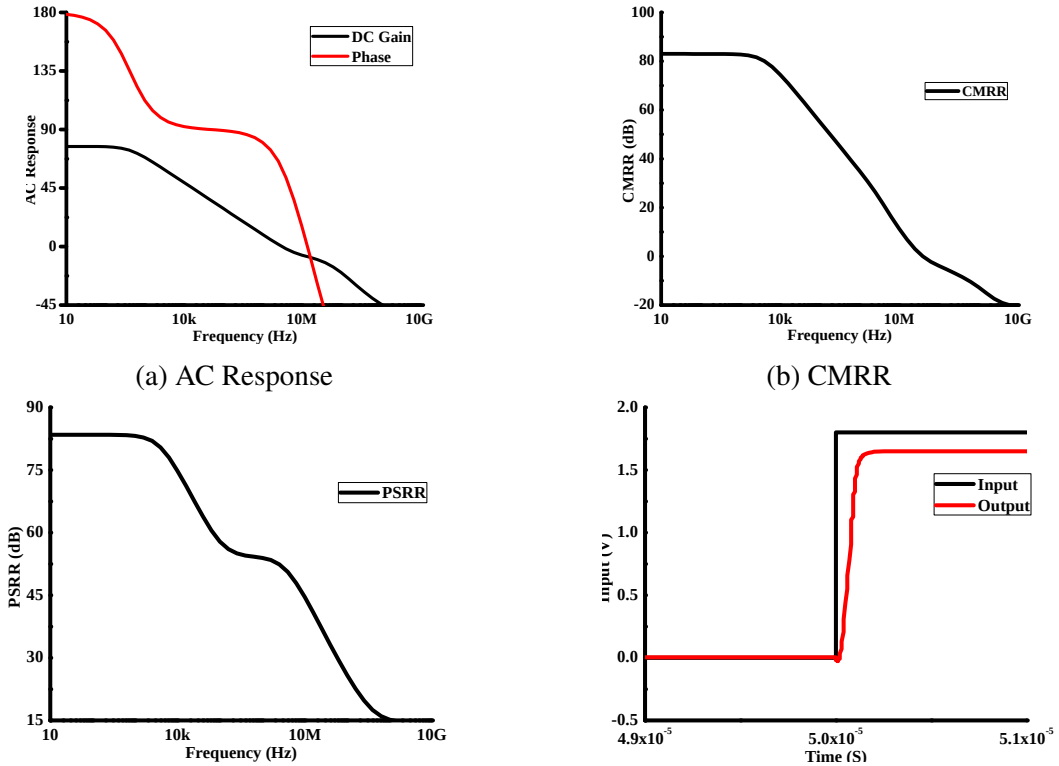
Design spec.	Target	PSO [119]	WOA [120]	PSO [120]	DE [120]	GSA [118]	AGSA [118]	WOA -mGWO
$A_v\ (dB)$	>60	59.19	74.08	61	74.086	60.14	81.13	75.3
$GBW\ (MHz)$	>3	3	3	3.85	3	3.136	3.29	3.32
$PM\ (degrees)$	>45	63.53	–	–	–	47.53	57.91	60.3
$SR\ (V/\mu s)$	>10	18.35	10	10	20	10.29	12.34	18.2
$Pd\ (mW)$	<2.5	0.184	1.137	0.979	2.5	1.053	0.332	0.09
$CL\ (pF)$	>7	–	7	7	7	10.02	10	7
$V_{ICmin}\ (V)$	>0.3	–	-0.01	-0.01	-0.01	-0.86	-1.22	0.4
$V_{ICmax}\ (V)$	<1.6	–	1.1	1.1	1.1	1.8	1.79	1.2
$CMRR\ (dB)$	>60	67.08	–	–	–	81.99	84.6	83.01
$PSRR+\ (dB)$	>70	63.84	–	–	–	77.36	97.45	84.3
$PSRR-\ (dB)$	>70	99.16	–	–	–	86.82	84.86	77.45
$Area(\mu m^2)$	Objective	28.52	93.86	148.2	93.867	114.6	70.32	21.51
FOM_1	Max.	–	0.43	1.283	0.43	0.69	1.09	1.225
FOM_2	Max.	–	0.197	0.186	0.01	0.26	1.41	24.32
Tech. (μm)		0.18	0.18	0.18	0.18	0.35	0.35	0.18

5.5.1 Statistical Study

The random nature of metaheuristic algorithms stops them from producing constant output for every execution. Therefore, executing these algorithms only once may not be appropriate to comment on their performance. Hence, a statistical study is required to verify the actual performance and robustness of the presented method.

The hybrid WOA-mGWO algorithm based optimization process is repeated for 20 times with 20 different initializations of search agents. Table 5.5 shows the actual performance of the proposed algorithm for its application to analog circuit sizing, with each column representing the results obtained by using the best, average and worst solutions from hybrid WOA-mGWO.

Moreover, it is observed that all individual runs lead to feasible solutions which demonstrates the strong exploration ability of hybrid WOA-mGWO based optimization process.



(a) AC Response

(b) CMRR

(c) PSRR

(d) Slew Rate

Figure 5.8: Response of two-stage CMOS OPAMP using WOA-mGWO.

Table 5.5: Statistical study of the results obtained by using WOA-mGWO for 20 independent runs.

Design Specifications	WOA-mGWO (best)	WOA-mGWO (median)	WOA-mGWO (worst)
A_v (dB)	75.30	75.68	67.04
$GBW(MHz)$	3.52	4.92	6.13
$PM(^\circ)$	60.50	54.75	67.35
$SR(V/\mu s)$	18.20	12.46	16.65
$Pd(\mu W)$	90	81	114
$CL(pF)$	7	7	7
$V_{ICmin}(V)$	0.4	0.6	0.6
$V_{ICmax}(V)$	1.4	1.4	1.52
$CMRR(dB)$	83.01	65.35	64.60
$PSRR+(dB)$	84.50	87.87	90.10
$PSRR^{-}(dB)$	77.45	67.51	71.70
$Area(\mu m^2)$	21.51	53.40	66.00

Table 5.6: Results obtained after performing corner analysis on best solution obtained using WOA-mGWO algorithm.

Cor.	Temp. (°)C	V_{dd} - 5% (V)			V_{dd} (V)			V_{dd} + 5% (V)		
		Av (dB)	GBW (MHz)	PM (°)	Av (dB)	GBW (MHz)	PM (°)	Av (dB)	GBW (MHz)	PM (°)
FF	-40	76.01	3.4	59.08	76.03	3.4	59.49	75.92	3.4	59.73
	25	75.4	3.4	59.29	75.78	3.4	59.88	75.59	3.58	58.6
	90	65.29	3.4	59.4	74.79	3.4	58.73	74.79	3.58	59.1
FS	-40	75.7	2.76	64.47	76	3.06	62.59	75.98	3.23	61.5
	25	60.2	2.62	65.36	75.44	3.23	61.31	75.48	3.4	60.31
	90	53.49	2.59	66.03	67.83	3.23	61.33	74.63	3.4	60.61
SS	-40	73.7	2.76	65.8	76.03	3.22	63.13	74.55	3.4	62.1
	25	56.53	2.66	67.09	75.17	3.23	62.59	75.5	3.4	61.9
	90	53.05	2.66	66.66	61.18	3.06	63.02	74.59	3.4	61.57
SF	-40	76.04	3.4	61.03	75.64	3.4	61.54	73.94	3.58	60.28
	25	74.65	3.4	60.55	75.81	3.58	59.94	74.55	3.58	60.26
	90	59.81	3.06	62.63	74.72	3.58	59.75	74.78	3.67	58.85
TT	-40	76.03	3.06	62.89	76.01	3.23	62.12	75.59	3.4	60.95
	25	67.46	3.06	62.49	75.3	3.52	60.5	75.35	3.58	59.58
	90	55.64	2.76	64.55	73.36	3.52	60.46	74.84	3.58	59.79

5.5.2 Monte-Carlo and Corner analysis

The corner analysis is usually performed to validate design efficiency and robustness, considering process corners, temperature and supply voltage variations. Here, the design with best solution obtained is examined for over 45 states as a result of cross combinations from five process corners i.e., FF, FS, SS, SF and TT, supply voltage variations i.e., $V_{dd} \pm 5\%$ and temperature variations i.e., -40°C, 25°C and 90°C. The numerical results of the corner analysis obtained for the design with the best solution obtained using WOA-mGWO are provided in Table 5.6. The values shown in Table 5.6 meet the specification requirements of the design for A_v, GBW and PM. Further stringent analysis for robustness of the proposed technique, considering mismatch and process variations, is performed using montecarlo simulation for over 2000 runs and the results are tabulated as illustrated in Table 5.7. The few of the important performance metrics considered include DC gain, GBW and phase margin that are observed to be almost sturdy at 73 dB, 3.50 MHz and 60 °, respectively.

Table 5.7: Simulation results from monte-carlo analysis (2000 runs).

Parameter	DC Gain (dB)	UGB (MHz)	Phase Margin (°)
Mean	73.1265	3.50249	60.52
Maximum	76.04	3.742	67.61
Minimum	48.18	2.591	56.79
Standard Deviation	4.89242	0.114786	1.78527

5.6 Summary

Over the past years, the automation of analog circuit design and sizing would have been more successful if the optimization algorithms had incorporated the major trade-offs like accuracy, robustness and run-time, simultaneously. This challenge is addressed by introducing a novel algorithm i.e., hybrid WOA-mGWO algorithm, considering it to be a step towards the improved performance of automated sizing tools. Here, the balance between exploration and exploitation is revisited to improve its exploration ability prior to convergence at globally optimal solution. The results obtained after evaluating a set of 23 benchmark functions is compared with algorithms from recent literature revealing its robustness and better performance owing to higher ability of exploration. Despite of the advantages, the hybrid WOA-mGWO algorithm suffers from slightly additional run-time due to inclusion of additional steps in the process of enhancement. The same is validated considering a two-stage CMOS OPAMP, in 180 nm CMOS technology as a benchmark circuit, resulting in a reduced area and power consumption, in comparison to the foregoing algorithms. To ensure the robustness of WOA-mGWO, a statistical study is performed with over 20 independent runs to return a feasible solution in every case. Besides, a statistical analysis, i.e., corner analysis and montecarlo simulation were performed to further evaluate the robustness of the design. Fine tuning the design parameters is one of the major challenges faced in this method which was overcome by manual tuning.

Chapter 6

Design Automation of CMOS Analog Circuits Using Novel Hybrid SCAWOA algorithm

6.1 Introduction

In the previous sections, the equation based methodology was discussed which suffers from accuracy issues. To overcome the disadvantages of the equation-based design methodology, a novel methodology is proposed using ocean scripts from cadence, UNIX shell and MATLAB. This chapter starts with a detailed explanation of Sine-Cosine Algorithm (SCA) and process of hybridization of SCA and WOA algorithm (SCAWOA) followed by evaluation of the hybrid SCAWOA algorithm by performing a comparative study over 23 classical benchmark functions. The proposed tool is validated using different analog circuits, i.e., folded cascode CMOS operational amplifier and capacitor-less low dropout regulator (LDO). The simulation results and their comparison with other circuit sizing tools is also presented.

6.2 Sine-Cosine Algorithm

Sine-cosine algorithm [117] is another population-based optimization algorithm, proposed by Seyedali Mirjalili, is based on the fluctuations outwards and towards the best solution based on mathematical models using sine and cosine functions. Here, the position of the search agents

is updated using the sine and cosine functions as shown in Equations 6.1 and 6.2 corresponding to exploitation and exploration.

$$X_i^{t+1} = X_i^t + r_1 sin(r_2) \left| r_3 P_i^t - X_i^t \right|, \quad r_4 < 0.5 \tag{6.1}$$

$$X_i^{t+1} = X_i^t + r_1 cos(r_2) \left| r_3 P_i^t - X_i^t \right|, \quad r_4 \geq 0.5 \tag{6.2}$$

Here, 'X_i^t' is the position of search agent at t^{th} iteration and i^{th} dimension, 'P_i' is the best solution in the i^{th} dimension, 'r_1', 'r_2' and 'r_3' are the random values, || represents the absolute value.

The parameter 'r_1' directs the movement of the search agent, i.e., the next position, that can be

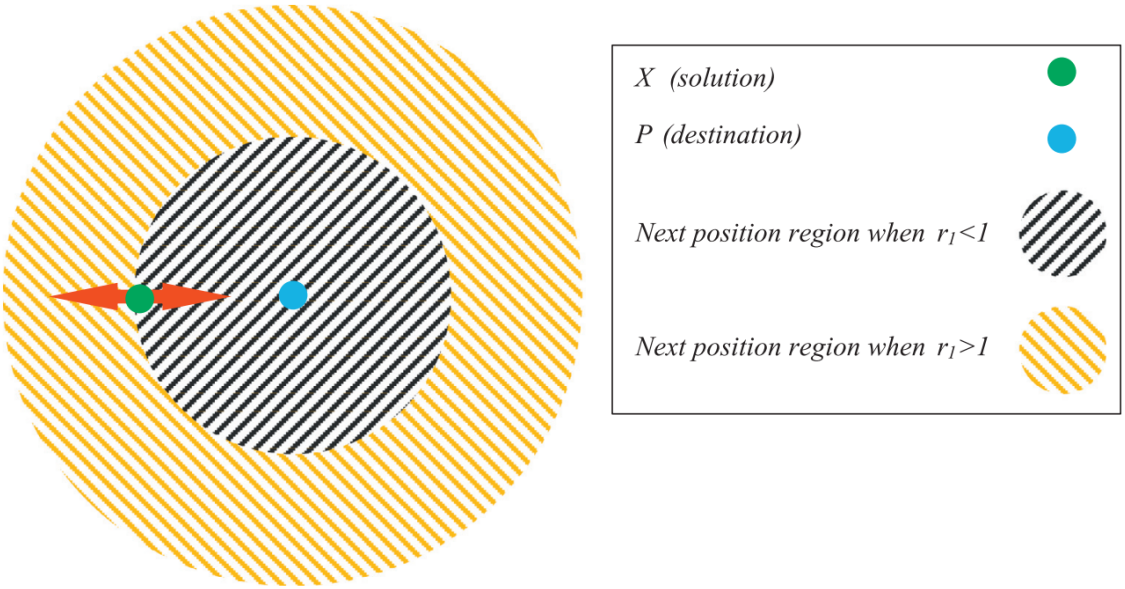


Figure 6.1: Effects of Sine and Cosine functions on the next position.[117]

anywhere between the destination and solution or outside it. The parameter 'r_2' is responsible to define the movement of the search agent outwards or towards the destination or best solution obtained so far. The variable 'r_3' emphasizes or de-emphasizes the effect of best solution or destination while defining the distance.

The dedication of the search agents to exploration and exploitation, while switching between sine and cosine functions, is decided by the random variable 'r_4' with the range [0,1] as shown in Equation 6.3.

$$X_i^{t+1} = \begin{cases} X_i^t + r_1 sin(r_2) \left| r_3 P_i^t - X_i^t \right| & r_4 < 0.5 \\ X_i^t + r_1 cos(r_2) \left| r_3 P_i^t - X_i^t \right| & r_4 \geq 0.5 \end{cases} \tag{6.3}$$

The graphical representation of the Equations 6.1 and 6.2 are demonstrated in Figure 6.1 It depicts the operation of the equations over two search agents in the entire search space.

The dimensions of the sine and cosine functions for maintaining the balance between

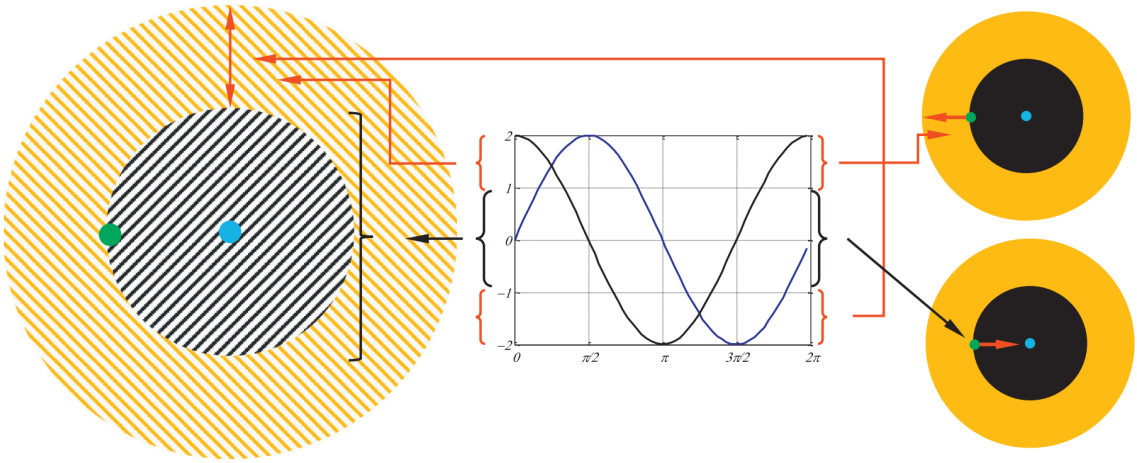


Figure 6.2: Sine and cosine with the range in $[-2, 2]$ allow a solution to go around or beyond the destination.[117]

exploration and exploitation are adaptively altered using Equation 6.4.

$$r_1 = a - t\frac{a}{T} \tag{6.4}$$

Where 'T' is maximum number of iterations, 'a' is constant defining the range of 'r_1' and 't' is the current iteration. Figure 6.2 demonstrates the theoretical model of sine cosine function within the range $[-a, a]$ with the value of $a = 2$. The pseudo code of the SCA algorithm is

Algorithm 3 Sine-cosine algorithm.

1: Initialize a set of search agents X_i (i = 1, 2, 3, 4, ... , n).
2: Calculate the fitness of each search agent.
3: X_p = the best search agent.
4: **while** (t< Maximum number of iterations)
5: Update $r_1, r_2, r_3 and r_4$.
6: Update the position of search agents using Equation 6.3
7: Update X_p if there is a better solution.
8: $t = t + 1$
9: **end while**
10: return X_p

represented in Algorithm 3. Here, the process starts by initialization of random search agents followed by the evaluation of the cost function. The best solution obtained so far is saved considering it as the target and updates the position of the search agents with respect to target till next best solution is obtained. Other parameters are updated to emphasize or de-emphasize

exploration and exploitation of entire search space with increase in the iterations. This process ends when the termination criteria is satisfied.

6.3 Hybrid of sine-cosine algorithm and whale optimization algorithm

WOA algorithm is one of the recently proposed optimization algorithm that shows superior results for many optimization problems. The exploitation in this algorithm is effected by the blind operator irrespective of the fitness of the current solution. This operator is replaced by the process used in SCA algorithm considering the positions of best search agent. This section discusses the detailed hybridization process of WOA with SCA. WOA is proven to be better for some cases by completely utilizing the population size with better performance on some of the benchmark functions. However, using WOA for complex optimization problems may lead to a non-optimal solution. For further enhancement of the exploitation in WOA, a more concentrated modified grey wolf optimization algorithm is combined with WOA to derive a new hybrid SCAWOA algorithm.

In hybrid SCAWOA, a standard WOA algorithm is used for globally searching the entire search space and to bring most of the solutions towards the favourable area. After the exploration phase, SCA algorithm is used to locally search for the best (optimal) solution. In Hybrid SCAWOA, the WOA algorithm emphasizes on the expansion at the starting point of search to explore the entire search space broadly, while later SCA algorithm focuses on the magnification by allowing the individuals to move towards the optimum solution as a next stage in the optimization process. This method is also capable of dealing with both global and local search processes adequately.

6.3.1 Steps of proposed Hybrid SCAWOA algorithm

The detailed flow of the proposed hybrid SCAWOA is presented below and the same is represented in the form of flowchart as shown in the Figure 6.3.

Step 1: Initializing the population size (N) randomly as a vector as a starting point of the optimization algorithm.

$$X_{i,j} = lb_j + rand()(ub_j - lb_j) \tag{6.5}$$

Where $X_(i, j)$ denotes the j^{th} dimension of i^{th} solution; ub_j and lb_j are the upper and lower bounds with j^th dimension; *rand*() takes any random number from zero to one.

Step 2: Evaluation of fitness function using each vector $X_{i,j}$ i.e., calculating the values of the function using $X_{i,j}$ for entire population and obtaining the best position.

Step 3: Initialization of algorithm parameters such as 'r_1', 'r_2', 'A', 'C', 'a', 'a_2', 'b' and 'p'. The variation of 'A_j' over the course of 500 iterations for different values of 'i' helped in determining the value of 'i' (= 3) for slightly increasing the number of iterations assigned for exploration.

$$a = 2(1 - \frac{l^3}{(Maxiter)^3}) \tag{6.6}$$

where 'l' is the current iteration and '$Maxiter$' is the maximum number of iterations.

Step 4: When $p < 0.5$ and $A < 1$, the position of the search agent is updated using equations shown below.

$$D_j = \left|C_j\, X_{jr}\,(t) - X_j\,(t)\right| \tag{6.7}$$

$$X_j\,(t+1\,) = \left|X_{jr}\,(t) - A_j\, D_j\right| \tag{6.8}$$

Step 5: When $p < 0.5$ and $A \geq 1$, the position update is done using following equations considering three best search agents.

$$D_{ja} = \left|C_{j1}\, X_{ja} - X_j\right|,\; D_{jb} = \left|C_{j2}\, X_{jb} - X_j\right|,\; D_{jd} = \left|C_{j3}\, X_{jd} - X_j\right| \tag{6.9}$$

$$X_{j1} = D_{aj} e^{bl} \cos\,(2\pi l\,) + X_{aj}\,(t)\,, \tag{6.10a}$$

$$X_{j2} = D_{bj} e^{bl} \cos\,(2\pi l\,) + X_{bj}\,(t)\,, \tag{6.10b}$$

$$X_{j3} = D_{dj} e^{bl} \cos\,(2\pi l\,) + X_{dj}\,(t) \tag{6.10c}$$

$$X_j\,(t+1) = \frac{\left(X_{j1} \;+\; X_{j2} \;+\; X_{j3}\right)}{3} \tag{6.11}$$

Step 6: When $p > 0.5$ and $r_4 < 0.5$ the position of the search agent is updated as follows.

$$X_i^{t+1} = X_i^t + r_1 sin(r_2)\left|r_3 P_i^t - X_i^t\right| \tag{6.12}$$

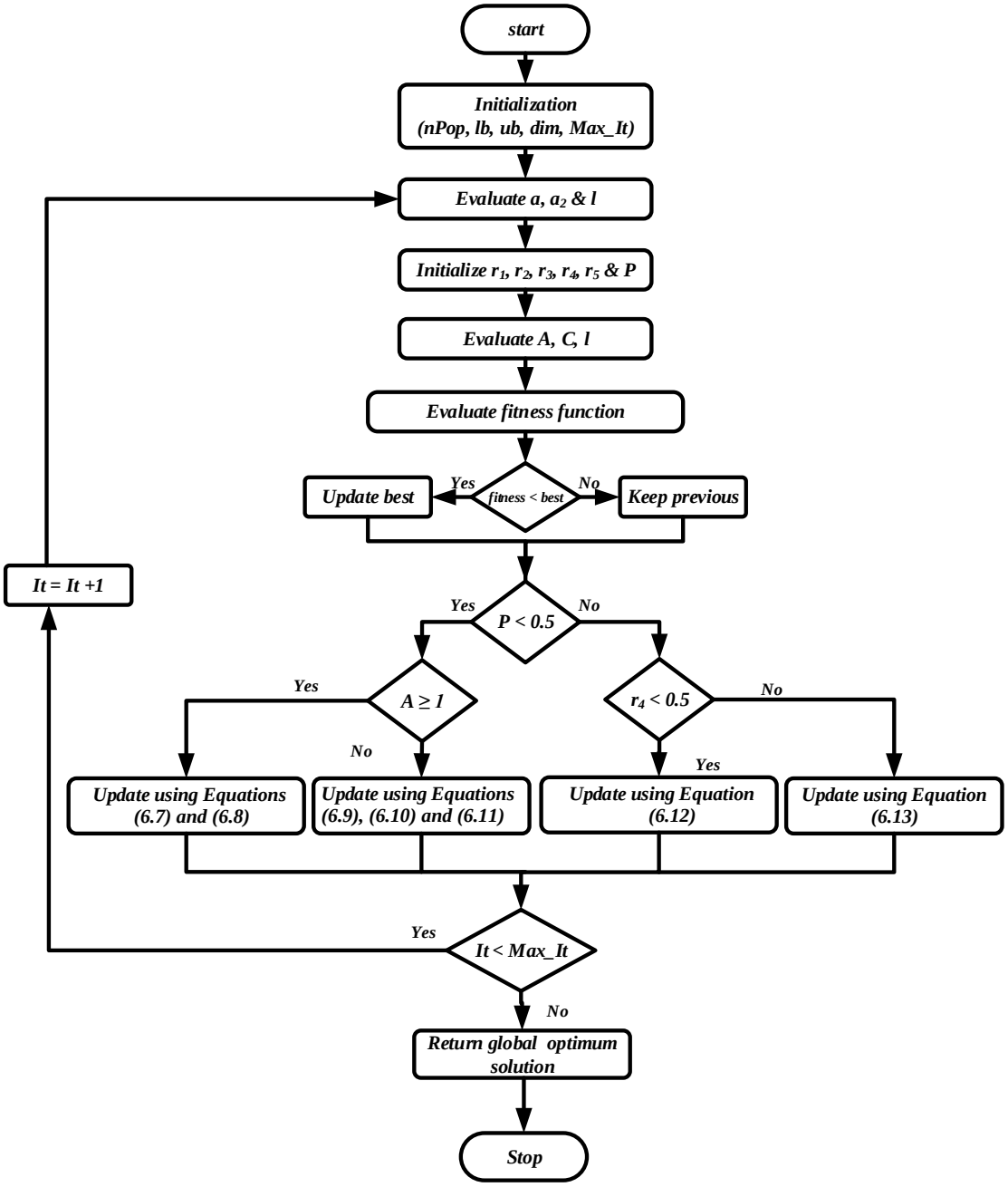


Figure 6.3: Optimization flow for hybrid SCAWOA algorithm.

Step 7: When $p > 0.5$ and $r_4 \geq 0.5$ the position of the search agent is updated as follows.

$$X_i^{t+1} = X_i^t + r_1 cos(r_2) \left| r_3 P_i^t - X_i^t \right| \tag{6.13}$$

Step 8: If any of the control variables violate the limits, then its value is set to lower or upper limit.

Step 9: Termination criterion is to be checked, i.e., whether the algorithm is run for maximum number of iterations specified.

Step 10: If termination criterion is not satisfied, go to Step 3. Else, print the optimum solution obtained.

6.4 Performance evaluation of hybrid SCAWOA algorithm

The performance of the hybrid SCAWOA is compared with other algorithms with respect to 23 classical and popular benchmark functions, to analyze the efficiency of the proposed algorithm. These benchmark functions [125], shown in Appendix A.2, are classified as unimodal, fixed (low) dimensional and multimodal high dimensional benchmark functions. The simulation results after evaluation of the said benchmark functions, with 20 independent runs, using the proposed hybrid SCAWOA algorithm are compared with other algorithms namely, SCA, WOA, PSO, GWO, Differential Evolution (DE) and GSAPSO. Table 6.1 reports Mean (M), Best (B), Worst (W) and Standard Deviation (SD) of the solutions obtained after the final iteration for 23 benchmark functions. Functions $F_1 - F_7$ are unimodal functions that have only global optimum solution. It can be observed from the results obtained for these unimodal functions that the hybrid SCAWOA outperforms other algorithms on four out of seven functions. The multimodal functions $F_8 - F_{13}$ have many local minima and the final results are more important as they demonstrate the algorithm's ability to avoid local optima. Testing on functions $F_8 - F_{13}$ show that on four out of six multimodal high-dimensional benchmark functions, proposed algorithm performs better than other algorithms. Functions $F_{14} - F_{23}$ are simpler due to a less number of local minimum solutions and low dimensionalities. Validation for this set of functions shows that hybrid SCAWOA outperforms other algorithms on six of ten of the multimodal low-dimensional benchmark functions. The convergence rates for functions $F_1 - F_{23}$ using GWO, SCA, WOA, PSO, GSAPSO and Differential Evolution (DE) algorithms have been investigated and are illustrated as shown in Figures 6.4 – 6.5. The descending trend proves the capability of the SCAWOA algorithm in obtaining solution over the course of iterations. Overall, these results show the potential of hybrid SCAWOA algorithm in solving problems that cannot be solved efficiently by other algorithms. Two metrics i.e., convergence rate and average fitness of search agents, are employed to verify the convergence of the proposed algorithm.

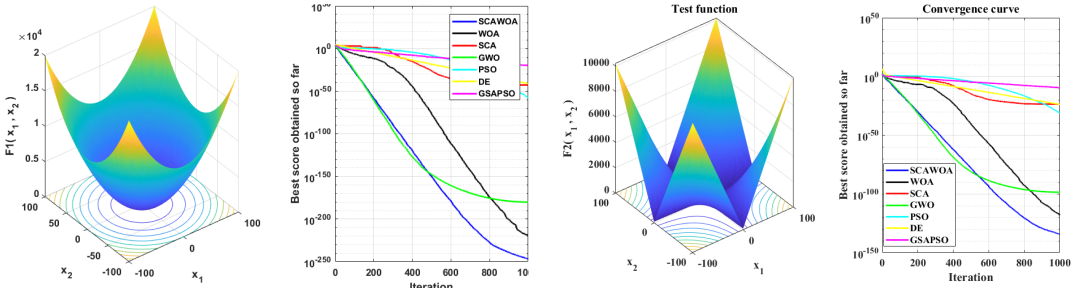
Table 6.1: Minimization results of 23 benchmark functions over 20 runs for $F1 - F23$.

F		SCAWOA	SCA	WOA	GWO	PSO	DE	GSAPSO
F_1	Best	**2.07E-201**	5.18E-40	8.60E-192	1.56E-149	5.71E-55	1.38E-43	5.91E-21
	Mean	**3.66E-193**	2.14E-30	4.56E-179	1.07E-142	1.45E-49	6.60E-42	1.25E-20
	SD	**0**	5.23E-30	0	3.90E-142	5.90E-49	1.35E-41	4.14E-21
	Median	**4.35E-198**	6.47E-33	2.00E-187	1.72E-144	3.79E-51	1.92E-42	1.23E-20
	Worst	**6.18E-192**	2.09E-29	9.11E-178	1.75E-141	2.65E-48	6.17E-41	2.28E-20
F_2	Best	**1.54E-123**	1.64E-25	4.40E-118	4.07E-84	4.10E-29	5.79E-25	1.37E-10
	Mean	**5.66E-111**	1.18E-19	5.54E-108	9.29E-81	1.16E-26	1.30E-24	2.33E-10
	SD	**2.49E-110**	5.23E-19	1.19E-107	2.83E-80	3.52E-26	6.84E-25	4.62E-11
	Median	**8.64E-116**	1.24E-22	6.60E-109	9.38E-82	1.03E-27	1.10E-24	2.40E-10
	Worst	**1.12E-109**	2.34E-18	4.47E-107	1.27E-79	1.56E-25	2.98E-24	3.17E-10
F_3	Best	1.54E-71	8.40E-20	3.59E-07	**9.12E-76**	3.53E-18	0.053351386	4.64E-21
	Mean	1.03E-50	7.01E-13	1.2299	**7.10E-68**	4.12E-15	0.233904838	2.53E-20
	SD	4.52E-50	1.69E-12	2.5161	**1.40E-67**	7.90E-15	0.149741928	1.13E-20
	Median	2.09E-58	9.48E-16	0.3054	**3.78E-71**	1.83E-16	0.170757692	2.53E-20
	Worst	2.02E-49	6.32E-12	8.9353	**5.51E-67**	2.96E-14	0.580543774	5.10E-20
F_4	Best	**5.52E-88**	4.69E-13	1.79E-08	5.40E-58	1.06E-14	7.88E-08	4.14E-11
	Mean	**8.96E-77**	3.30E-10	0.0914	1.38E-45	4.91E-13	1.69E-07	6.26E-11
	SD	**3.63E-76**	4.50E-10	0.2939	4.78E-45	8.12E-13	4.76E-08	9.31E-12
	Median	**1.19E-79**	7.19E-11	0.0002	7.47E-47	9.48E-14	1.70E-07	6.21E-11
	Worst	**1.63E-65**	1.61E-09	1.2634	2.16E-44	3.35E-12	2.64E-07	7.87E-11
F_5	Best	5.3359	6.2345	4.9999	5.0873	**0.0063**	0.1156	0.6962
	Mean	5.9676	7.1149	5.7743	5.9619	3.4496	**2.3504**	3.6918
	SD	**0.2198**	0.4547	0.3232	0.5986	2.4192	1.8371	4.2474
	Median	6.0449	7.2102	5.8304	6.1341	4.0003	2.3131	**1.3779**
	Worst	6.2331	8.0607	6.2420	7.1910	9.0915	**5.9200**	15.018
F_6	Best	3.56E-06	0.0961	1.21E-06	3.01E-07	**0**	0	6.24E-21
	Mean	1.14E-05	0.3062	9.39E-06	6.89E-07	**0**	0	1.22E-20
	SD	5.98E-06	0.1654	9.16E-06	2.64E-07	**0**	0	4.01E-21
	Median	1.13E-05	0.2999	7.55E-06	6.66E-07	**0**	0	1.20E-20
	Worst	1.98E-05	6.24E-01	3.63E-05	1.27E-06	**0**	0	2.06E-20
F_7	Best	**9.17E-07**	4.23E-05	1.01E-05	6.46E-05	0.0006	0.0008	0.0009
	Mean	**0.0001**	0.0007	0.0009	0.0002	0.0021	0.0023	0.0045
	SD	**9.60E-05**	0.0006	0.0011	0.0001	0.0013	0.0011	0.0027
	Median	**0.0001**	0.0005	0.0003	0.0001	0.0020	0.0021	0.0041
	Worst	**0.0004**	0.0026	0.0039	0.0006	0.0059	0.0046	0.0108
F_8	Best	-4189.8	-2427.3	**-4189.8**	-3419.9	-3538.4	-4189.8	-3676.4
	Mean	-3235.6	-2257.0	-3627.7	-2882.8	-2644.9	**-4183.9**	-3108.5
	SD	646.04	112.29	661.94	335.86	436.02	**26.483**	302.75
	Median	-3005.2	-2290.7	-4010.4	-2900.5	-2640.1	**-4189.8**	-3111.7
	Worst	-2348.1	-2057.8	-2447.4	-2313.8	-1910.8	**-4071.3**	-2569.6

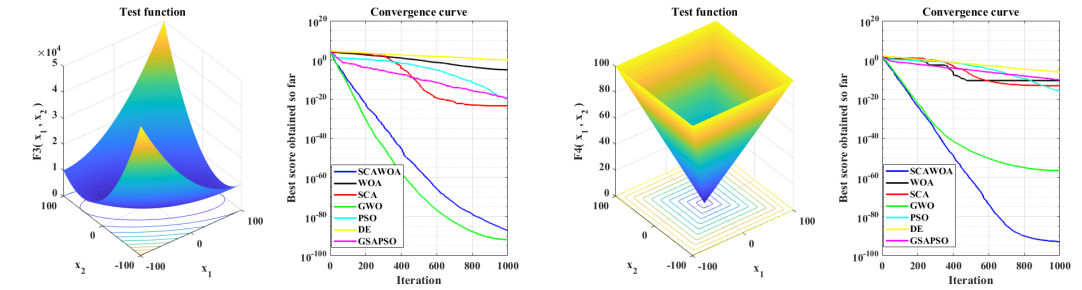
F		SCAWOA	SCA	WOA	GWO	PSO	DE	GSAPSO
F_9	Best	**0**	0	0	0	0.9949	0	4.9748
	Mean	**0**	0.9672	0	0.1570	3.0346	0	29.3014
	SD	**0**	4.3257	0	0.7023	1.5970	0	15.0147
	Median	**0**	0	0	0	2.9848	0	28.8537
	Worst	**0**	1.93E+01	0	3.1407	5.9698	0	54.722
F_{10}	Best	**8.88E-16**	8.88E-16	8.88E-16	4.44E-15	4.44E-15	4.44E-15	9.30E-11
	Mean	**3.16E-15**	4.44E-15	3.20E-15	4.97E-15	4.80E-15	4.44E-15	1.26E-10
	SD	**7.94E-16**	1.15E-15	2.38E-15	1.30E-15	1.09E-15	0	1.68E-11
	Median	**4.44E-15**	4.44E-15	4.44E-15	4.44E-15	4.44E-15	4.44E-15	1.27E-10
	Worst	**4.44E-15**	7.99E-15	7.99E-15	7.99E-15	7.99E-15	4.44E-15	1.52E-10
$F11$	Best	**0**	0	0	0	0.0443	0	0.0688
	Mean	**0**	0.0377	0.0395	0.0131	0.1497	0	0.1552
	SD	**0**	0.1379	0.0957	0.0220	0.0991	0	0.0929
	Median	**0**	0	0	0	0.1440	0	0.1206
	Worst	**0**	0.6065	0.3774	0.0715	0.4546	0	0.3936
F_{12}	Best	5.90E-06	0.0327	2.09E-06	7.43E-08	4.71E-32	**4.71E-32**	2.80E-22
	Mean	0.0045	0.0709	2.56E-05	2.33E-07	4.73E-32	**4.71E-32**	0.2800
	SD	0.0074	0.0380	3.34E-05	1.28E-07	3.92E-34	**5.62E-48**	0.5037
	Median	2.80E-05	0.0617	1.37E-05	2.22E-07	4.71E-32	**4.71E-32**	6.05E-22
	Worst	0.0197	0.2002	0.0001	5.77E-07	4.81E-32	**4.71E-32**	1.5573
F_{13}	Best	1.75E-05	0.0792	1.65E-05	4.62E-07	1.35E-32	**1.35E-32**	9.58E-22
	Mean	0.0375	0.2315	0.0011	0.0149	1.35E-32	**1.35E-32**	2.97E-21
	SD	0.0545	0.0673	0.0033	0.0364	2.81E-48	**2.81E-48**	9.75E-22
	Median	0.0002	0.2335	5.16E-05	8.15E-07	1.35E-32	**1.35E-32**	2.93E-21
	Worst	0.1971	0.3467	0.0110	0.1011	1.35E-32	**1.35E-32**	4.93E-21
F_{14}	Best	**0.9980**	0.9980	0.9980	0.9980	0.9980	0.9980	0.9980
	Mean	**0.9980**	1.2956	1.2956	2.6689	2.184744569	0.9980	2.5642
	SD	**0**	0.7268	0.72687	2.9238	1.7110	0	3.3796
	Median	**0.9980**	0.9980	0.9980	0.9980	1.9920	0.9980	0.9980
	Worst	**0.9980**	2.9821	2.9821	10.7631	5.9288	0.9980	15.5038
F_{15}	Best	**0.0003**	0.0003	0.0003	0.0003	0.0003	0.0004	0.0003
	Mean	**0.0004**	0.0008	0.0005	0.0014	0.0007	0.0006	0.0054
	SD	**5.53E-05**	0.0004	0.0003	0.0044	0.0002	8.57E-05	0.0088
	Median	**0.0003**	0.0007	0.0003	0.0003	0.0008	0.0006	0.0003
	Worst	**0.0006**	0.0014	0.0014	0.0203	0.0010	0.0007	0.0203
F_{16}	Best	**-1.0316**	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316
	Mean	**-1.0316**	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316
	SD	**1.36E-16**	1.23E-05	3.25E-12	2.48E-09	2.28E-16	2.28E-16	2.10E-16
	Median	**-1.0316**	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316
	Worst	**-1.0316**	-1.0315	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316

F		SCAWOA	SCA	WOA	GWO	PSO	DE	GSAPSO
F_{17}	Best	**0.3978**	0.3979	0.3978	0.3978	0.3978	0.3978	0.3978
	Mean	**0.3978**	0.3985	0.3978	0.3978	0.3978	0.3978	0.3978
	SD	**0**	0.0007	1.77E-07	9.97E-07	0	0	3.15E-06
	Median	**0.3978**	0.3983	0.3978	0.3978	0.3978	0.3978	0.3978
	Worst	**0.3978**	0.4015	0.3978	0.3978	0.3978	0.3978	0.3978
F_{18}	Best	**3**	3	3	3	3	3	3
	Mean	**3**	3	3	3	3	3	3
	SD	**3.47E-16**	7.03E-06	3.15E-06	8.04E-06	3.95E-16	5.49E-16	1.34E-15
	Median	**3**	3	3	3	3	3	3
	Worst	**3**	3	3	3	3	3	3
F_{19}	Best	-3.8627	-3.8622	-3.8627	-3.86278	-3.8627	-3.8627	**-3.8627**
	Mean	-3.8625	-3.8570	-3.8619	-3.8609	-3.8627	-3.8627	**-3.8627**
	SD	0.0002	0.0034	0.0014	0.0032	2.28E-15	2.28E-15	**2.22E-15**
	Median	-3.8626	-3.8548	-3.8625	-3.8627	-3.8627	-3.8627	**-3.8627**
	Worst	-3.8618	-3.8534	-3.8574	-3.8549	-3.8627	-3.8627	**-3.8627**
F_{20}	Best	**-3.3220**	-3.1247	-3.3219	-3.3219	-3.3219	-3.3219	-3.3219
	Mean	**-3.3220**	-3.0491	-3.2661	-3.2630	-3.2625	-3.3219	-3.2803
	SD	**2.07E-06**	0.0504	0.0717	0.0767	0.0609	5.06E-06	0.0581
	Median	**-3.3220**	-3.0138	-3.3218	-3.3219	-3.2625	-3.3219	-3.3219
	Worst	**-3.3220**	-3.0054	-3.1215	-3.1326	-3.2031	-3.3219	-3.2031
F_{21}	Best	-10.1531	-6.7229	-10.1531	-10.1531	-10.1531	**-10.1531**	-10.1531
	Mean	-9.3878	-3.6030	-8.8778	-9.3905	-8.1404	**-10.1531**	-5.1267
	SD	1.8673	2.2983	2.2644	1.8620	2.8928	**3.18E-15**	2.8087
	Median	-10.1525	-4.6884	-10.1526	-10.1529	-10.1531	**-10.1531**	-5.0551
	Worst	-5.0551	-0.4972	-5.0543	-5.0551	-2.6304	**-10.1531**	-2.6304
F_{22}	Best	**-10.4029**	-8.5410	-10.4029	-10.4028	-10.4029	-10.4029	-10.4029
	Mean	**-10.4029**	-4.4858	-8.4727	-10.1368	-9.8755	-10.4029	-4.6916
	SD	**3.35E-12**	1.6657	2.7133	1.1884	1.6233	5.65E-12	3.0641
	Median	**-10.4029**	-4.9030	-10.4024	-10.4026	-10.4029	-10.4029	-2.7658
	Worst	**-10.4029**	-0.9066	-3.7242	-5.0876	-5.1288	-10.4029	-1.8375
F_{23}	Best	-10.5364	-7.1793	-10.536	-10.5363	-10.5364	**-10.5364**	-10.5364
	Mean	-9.9946	-4.4619	-9.6084	-10.5361	-9.7271	**-10.5364**	-6.0070
	SD	1.6642	1.3809	2.3058	0.0001	1.9752	**5.21E-10**	3.8550
	Median	-10.5355	-4.8778	-10.5358	-10.5361	-10.5364	**-10.5364**	-3.8354
	Worst	-5.1285	-0.9464	-2.8065	-10.5356	-5.1284	**-10.5364**	-1.6765

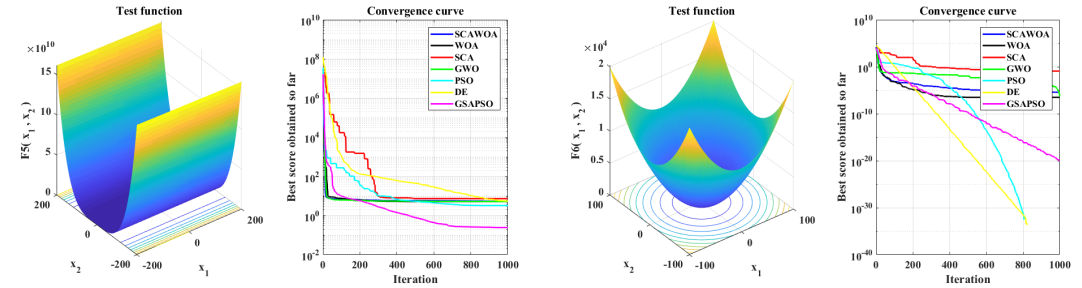
The descending trend demonstrates the ability of hybrid SCAWOA algorithm in obtaining a better global optimum solution over the course of iterations. Overall, these results show the potential of hybrid SCAWOA through convergence plots prove the efficiency of Hybrid SCA-WOA in avoiding the local minima with superior convergence rate for multimodal functions.



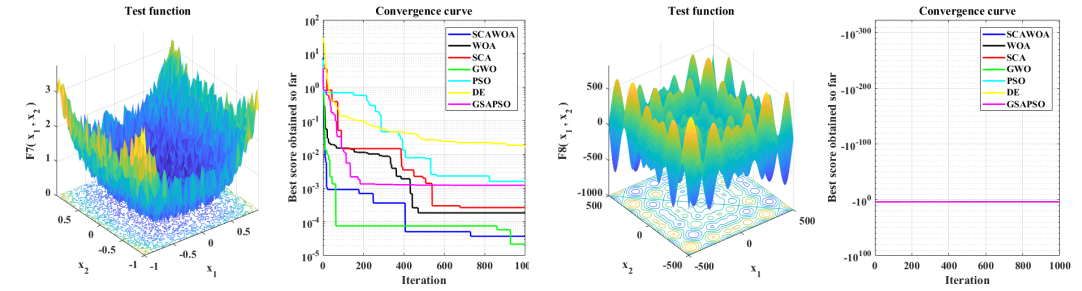
(a) F_1 (b) F_2



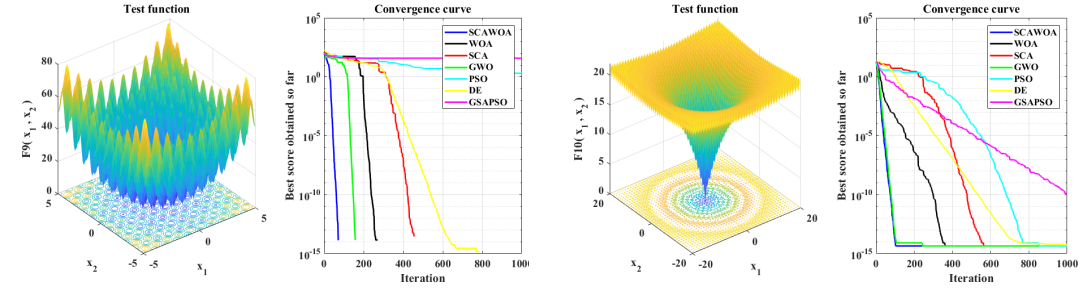
(c) F_3 (d) F_4



(e) F_5 (f) F_6

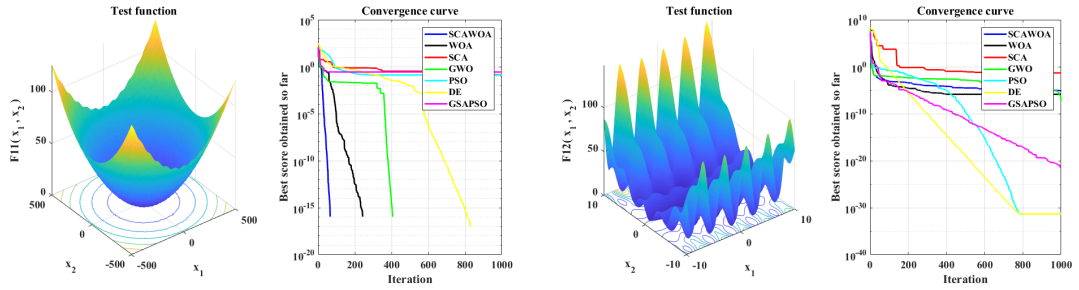


(g) F_7 (h) F_8

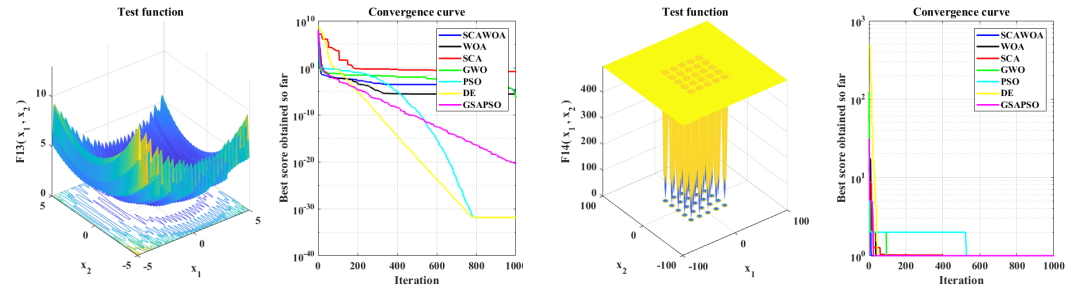


(i) F_9 (j) F_{10}

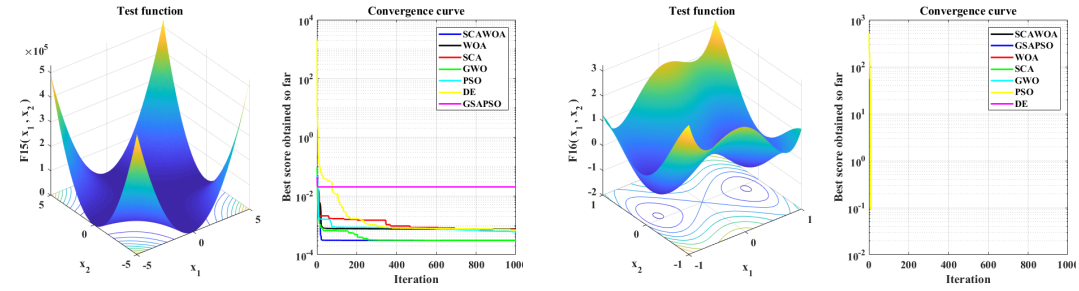
Figure 6.4: Convergence plots of benchmark functions (F_1 - F_{10}) for 3000 iterations.



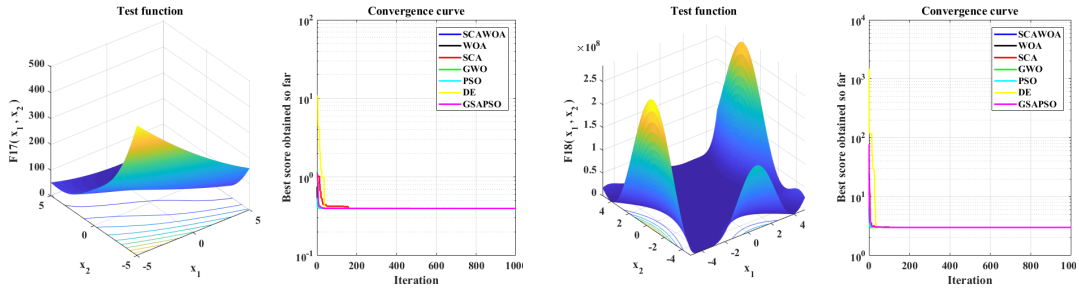
(a) F_{11} (b) F_{12}



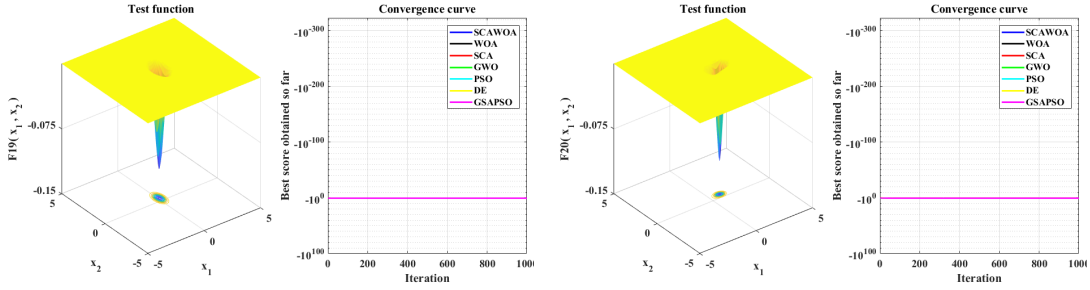
(c) F_{13} (d) F_{14}



(e) F_{15} (f) F_{16}

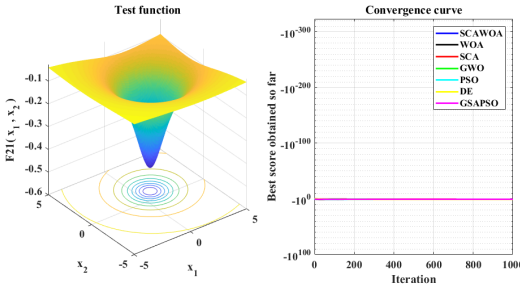


(g) F_{17} (h) F_{18}

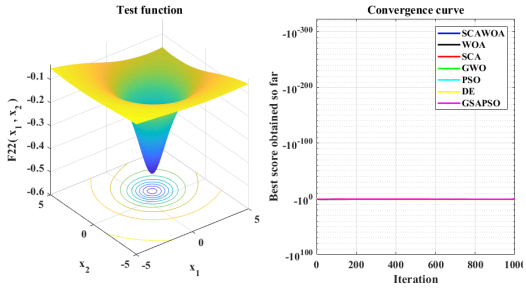


(i) F_{19} (j) F_{20}

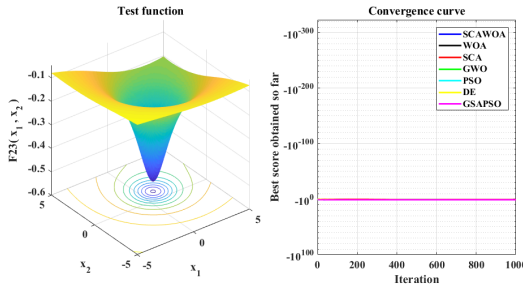
Figure 6.5: Convergence plots of benchmark functions ($F_1$1 - F_{20}) for 3000 iterations.



(a) F_{21}



(b) F_{22}



(c) F_{23}

Figure 6.6: Convergence plots of benchmark functions (F_{21} - F_{23}) for 3000 iterations.

Hence, it can be stated that the SCAWOA is good in maintaining proper balance between exploration and exploitation for solving various problems.

Comparison of the algorithms based on best, mean, worst and standard deviation values over 20 independent runs cannot be compared for each of the individual runs. Hence, there is always a possibility that the predominance may have occurred by chance despite its low probability in 20 runs. The Wilcoxon statistical test is performed at 5 percent significance level and the p-values are compared for each run and the significance of the results is decided, as reported in Table 6.2. For the statistical test, the best algorithm in each test function is compared with other algorithms independently. For example, if the best algorithm is SCAWOA, a pairwise comparison is made between SCAWOA/SCA, SCAWOA/GWO, SCAWOA/PSO, and so on. Since the best algorithm cannot be compared with itself, the corresponding value is replaced with NA for the best algorithm which stands for Not Applicable.

As shown in Table 6.2, p-values for most of the functions are much less than 5% for the SCAWOA, which demonstrates its statistical significance. For function F_{21}, the SCAWOA algorithm provides p-value greater than 0.05 that the SCAWOA algorithm is not better for this function which proves the NFL theorem. Overall, these results show that SCAWOA can outperform competing algorithms.

Table 6.2: P-values calculated for the Wilcoxon rank-sum test

	SCAWOA	SCA	WOA	GWO	PSO	DE	GSAPSO
F_1	NA	6.80E-08	6.80E-08	6.80E-08	6.80E-08	6.80E-08	3.65E-05
F_2	NA	6.80E-08	6.80E-08	6.80E-08	6.80E-08	6.80E-08	3.63E-05
F_3	NA	0.7608	3.66E-05	NA	3.66E-05	3.66E-05	3.66E-05
F_4	NA	6.80E-08	6.80E-08	6.80E-08	6.80E-08	6.80E-08	3.66E-05
F_5	2.72E-04	1.52E-04	2.47E-04	2.25E-04	0.487	NA	5.2E-04
F_6	NA	1.31E-07	1.31E-07	1.31E-07	NA	NA	1.31E-07
F_7	NA	1.41E-05	2.36E-06	0.0015	6.80E-08	6.80E-08	3.66E-05
F_8	3.80E-09	3.80E-09	3.80E-09	3.80E-09	3.80E-09	NA	3.80E-09
F_9	NA	NA	NA	NA	5.76E-09	NA	1.31E-07
F_{10}	NA	0.163	0.957	0.0393	0.0398	0.039	2.23E-05
F_{11}	NA	0.169	0.116	0.024	3.53E-07	0.162	4.11E-07
F_{12}	1.35E-05	1.35E-05	1.35E-05	1.35E-05	0.103	NA	1.35E-05
F_{13}	1.35E-05	1.35E-05	1.35E-05	1.35E-05	0.1036	NA	1.35E-05
F_{14}	NA	0.016	5.17E-06	0.0638	6.02E-05	8.01E-09	4.9E-03
F_{15}	NA	6.80E-08	6.80E-08	0.0002	6.80E-08	6.80E-08	3.61E-05
F_{16}	NA	6.80E-08	6.80E-08	0.4407	8.01E-09	8.01E-09	0.0199
F_{17}	NA	6.80E-08	1.05E-06	0.818	8.01E-09	8.01E-09	3.79E-09
F_{18}	NA	0.0012	1.20E-06	0.001	1.13E-08	3.93E-08	3.80E-09
F_{19}	NA	0.0845	2.28E-08	0.1162	5.20E-04	1.35E-05	NA
F_{20}	NA	6.80E-08	0.0638	0.0105	0.0289	1.13E-08	1.26E-05
F_{21}	3.80E-09	3.80E-09	3.80E-09	3.80E-09	0.0043	NA	0.1036
F_{22}	NA	0.0001	0.0001	0.0005	0.0016	4.21E-08	2.75E-05
F_{23}	3.80E-09	3.80E-09	3.80E-09	3.80E-09	0.0561	NA	0.1036

6.5 Application of hybrid SCAWOA algorithm for the design of two-stage CMOS operational amplifier

This section discusses the circuit-level implementation and the optimization of the design parameters for analog circuits employing the hybrid SCAWOA algorithm. The formulation of the cost functions for evaluating he efficiency of the proposed algorithms is discussed in Section 5.4. The objectives considered for optimization are MOS transistor area and power consumption. The design parameters of a two-stage CMOS OPAMP obtained using hybrid SCAWOA and other related algorithms are presented in the Table 6.3. Table 6.4 demonstrates the performance of SCAWOA algorithm for the design of two-stage CMOS operational amplifier and its comparison with other competing algorithms. The proposed technique results in low power making it relatively preferable for low power circuit sizing problem. For a fair comparison between recent techniques using algorithms such as PSO, WOA, DE, GSA, AGSA, and GWO, with application to circuit sizing that utilize different technologies, two Figures of

Table 6.3: Design parameters obtained using hybrid SCAWOA and its comparison using different algorithms.

Design parameters	Simulation-based		Equation-based				
	GSA [118]	AGSA [118]	PSO [119]	WOA [120]	PSO [120]	DE [120]	SCA-WOA
W_1/L_1	4/2	4/2	7.74/0.18	4/2	4/2	4/2	6/0.5
W_3/L_3	4/2	4/2	14.4/0.18	5/2	4/2	4/2	3/0.5
W_5/L_5	4/2	4/2	1.96/0.18	2/2	4/2	2/2	3.75/0.5
W_6/L_6	21.94/2	7.16/2	98.23/0.18	21.54/2	33.4/2	21.5/2	6.5/0.5
W_7/L_7	11.36/2	4/2	11.97/0.18	5.38/2	16.7/2	5.38/2	3.75/0.5
W_8/L_8	4/2	4/2	1.96/0.18	2/2	16.7/2	5.38/2	3.75/0.5
I_{bias} (μA)	45.28	30	34.46	49	21	49	20
C_C (pF)	4.4	2.2	–	2.45	2.45	2.45	2.15

Units of W_i and L_i are μm, where i = 1 to 8

Merit(FOMs) are considered which are given in Equation 3.19 and Equation 3.20. As can be observed from the Table 6.4, the presented hybrid SCAWOA results in better performance in circuit sizing, aiming at reduction in MOS transistor area while minimizing overall power consumption, when compared to the aforementioned algorithms.

Table 6.4: Results obtained using hybrid SCAWOA algorithm and their comparison with different algorithms.

Design spec.	Target	PSO [119]	WOA [120]	PSO [120]	DE [120]	GSA [118]	AGSA [118]	SCA WOA
A_v (dB)	>60	59.19	74.08	61	74.086	60.14	81.13	78.25
GBW (MHz)	>3	3	3	3.85	3	3.136	3.29	6.3
PM $(degrees)$	>45	63.53	–	–	–	47.53	57.91	59.63
SR $(V/\mu s)$	>10	18.35	10	10	20	10.29	12.34	10.85
Pd (mW)	<2.5	0.184	1.137	0.979	2.5	1.053	0.332	0.071
CL (pF)	>7	–	7	7	7	10.02	10	7
V_{ICmin} (V)	>0.3	–	-0.01	-0.01	-0.01	-0.86	-1.22	0.5
V_{ICmax} (V)	<1.6	–	1.1	1.1	1.1	1.8	1.79	1.2
$CMRR$ (dB)	>60	67.08	–	–	–	81.99	84.6	79.93
$PSRR+$ (dB)	>70	63.84	–	–	–	77.36	97.45	85.12
$PSRR-$ (dB)	>70	99.16	–	–	–	86.82	84.86	81.61
$Area(\mu m^2)$	Objective	28.52	93.86	148.2	93.867	114.6	70.32	19.31
FOM_1	Max.	–	0.43	1.283	0.43	0.69	1.09	2.205
FOM_2	Max.	–	0.197	0.186	0.01	0.26	1.41	32.17
Tech. (μm)		0.18	0.18	0.18	0.18	0.35	0.35	0.18

6.6 The proposed tool for analog IC sizing using hybrid SCAWOA algorithm

In this section, a novel sizing tool is proposed for automated design of analog circuits. Basically, circuit sizing tools consist of synthesis and optimization sections that are linked together. Here, The synthesis section uses the simulation-based optimization method, and the optimization section employs the proposed algorithm, hybrid SCAWOA. The analog circuits are simulated using SPECTRE simulator with hybrid SCAWOA as the optimization engine. Both the sections of this tool are connected together through the link between MATLAB and CADENCE.

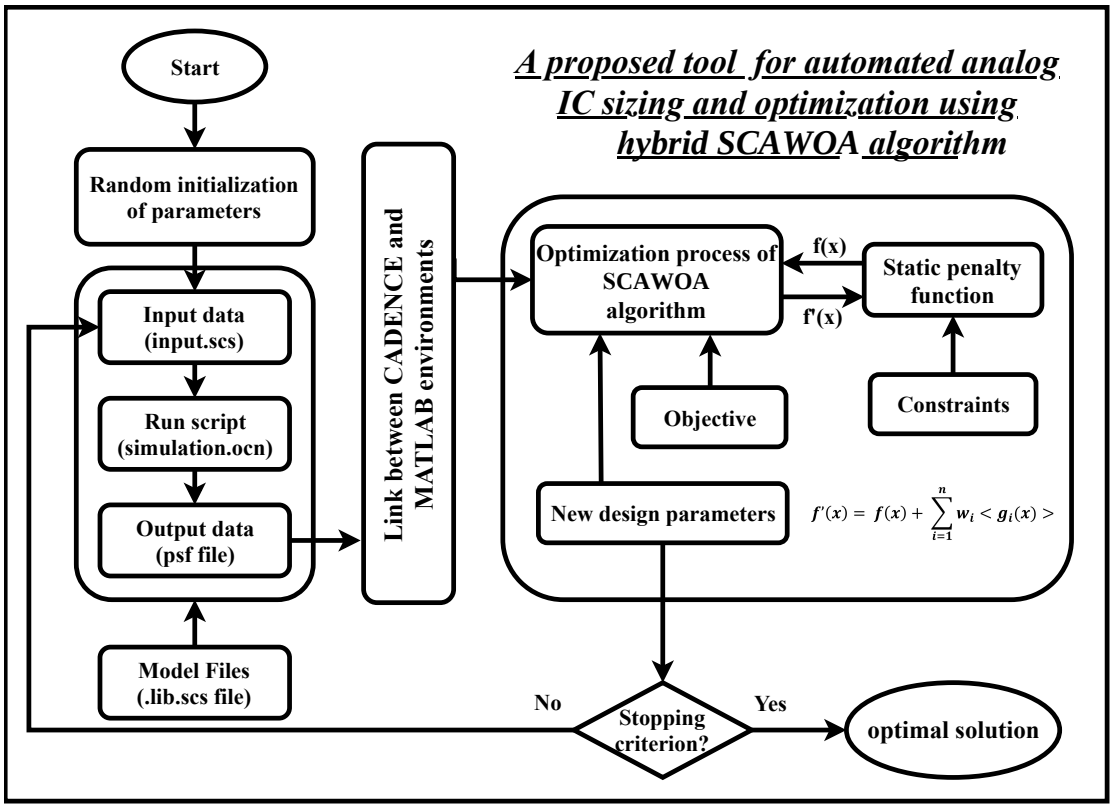


Figure 6.7: Simulation-based circuit sizing approach using SACWOA algorithm.

6.6.1 Architecture of the proposed tool

This section describes the mechanism of proposed tool in analog circuit sizing and optimization. The architecture of this tool as illustrated in Figure 6.7 indicates that the designer

provides the design parameters and constraints, at the beginning, while simultaneously defining the range for each design parameter to setup the search space.The design parameters include biasing voltages and currents, aspect ratios of MOS transistors and capacitance values. The process starts by initializing the randomly generated population of design parameters then supplying it to the synthesis section for starting the design process.

In synthesis section, the input spectre file, i.e., .scs file, contains the initial parameters that are simulated in cadence and saves the desired performance specifications as an output file. The proposed tool reads the output file through the link between cadence and MATLAB and evaluates the cost function while handling the performance and design constraints. Further, new design parameters are created by the tool for the next iteration using the optimization engine. This process is continued until the stop criteria are satisfied. The stopping criteria include maximum number of iterations and reaching the same value of objective for specific number of iterations. Finally, when the optimal circuit sizes are found, they are reported to the designer.

6.6.2 Case study: Folded cascode CMOS operational transconductance amplifier

The performance of the proposed tool in the analog circuit sizing is validated using the design of a CMOS FCOTA as a case study (See Figure 6.8).

The aim of the proposed tool is to minimize the total CMOS transistor area while satisfying the performance and functional constraints of maintaining all the transistors in the saturation region. The total MOS transistor area and power consumption are considered as the cost functions which are represented as shown below.

$$CF_1 = \sum_{i=0}^{N} W_i \times L_i \tag{6.14}$$

$$CF_2 = I_{total} \times V_{DD} \tag{6.15}$$

$$CF = CF_1 + CF_2 \tag{6.16}$$

Table 6.5 illustrates the optimal design parameters of the FCOTA obtained using the SCAWOA-based tool. The design specifications of the FCOTA including SR, A_v, PM, CMRR, and PSRR are plotted by the proposed tool, in Figure 6.9 to Figure 6.12 , for different technologies i.e., 180 nm (V_{DD} = 1.8 V, 3.3 V), 130 nm and 65 nm. Different feasible solutions are obtained at

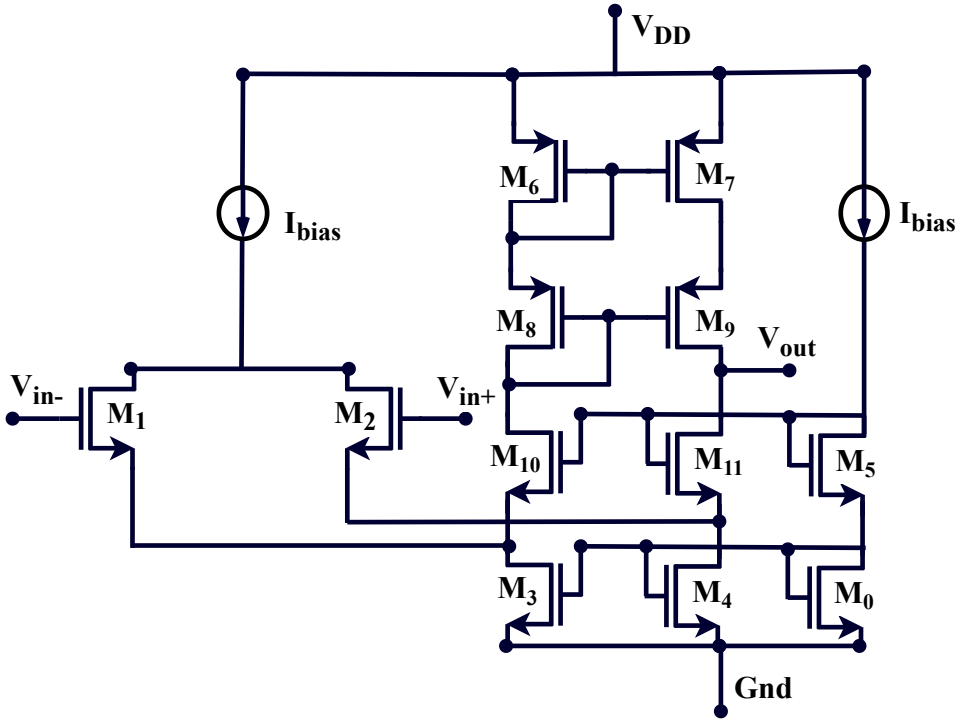
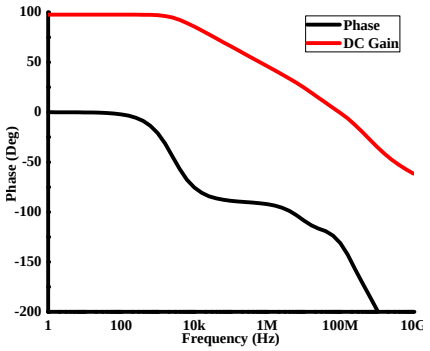
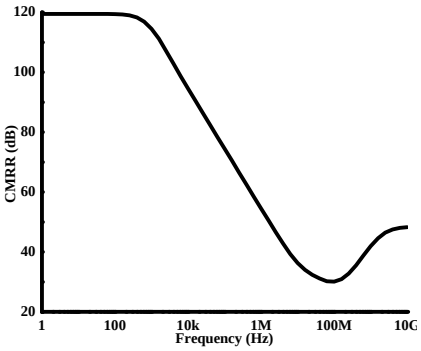


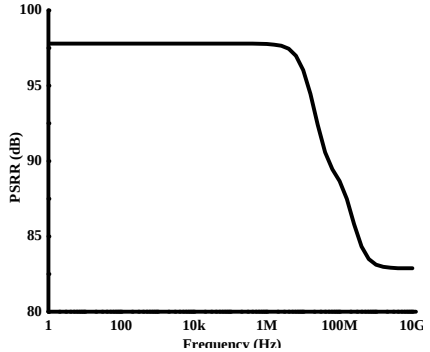
Figure 6.8: Schematic of FCOTA.



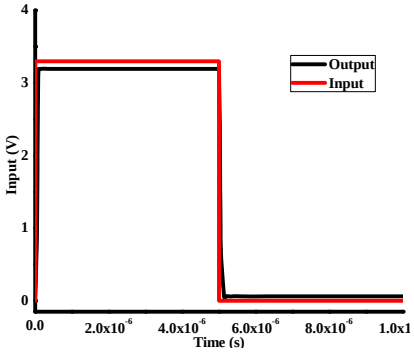
(a) AC Response



(b) CMRR



(c) PSRR

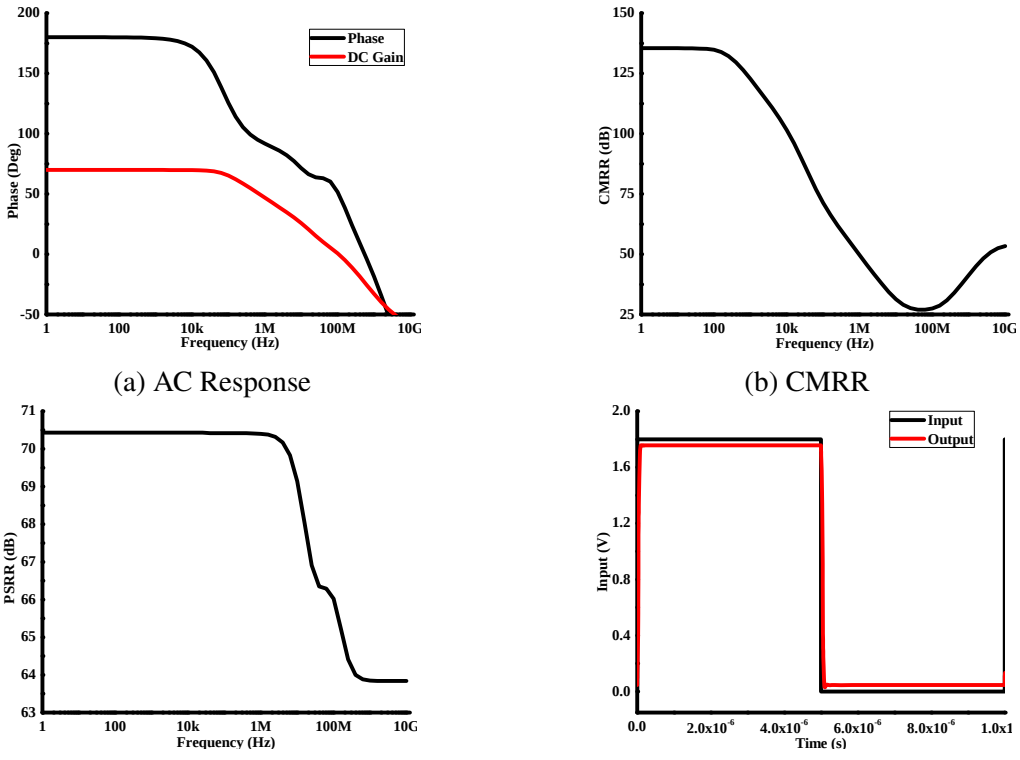


(d) Slew Rate

Figure 6.9: Simulation results for FCOTA in CMOS 180 nm technology ($V_{dd} = 3.3V$).

Table 6.5: Design parameters of FCOTA for different technologies obtained using simulation-based design methodology.

Parameter	FCOTA			
Tech.	180 nm		130 nm	65 nm
V_{DD} (V)	3.3	1.8	1.2	1.2
W_0 (μm)	1	2.95	1	2.852
W_1 (μm)	1	1.46	1	2.296
W_2 (μm)	1	1.46	1	2.296
W_3 (μm)	1	3.1	1	1.422
W_4 (μm)	1	3.1	1	1.422
W_5 (μm)	1	2.94	1	9.986
W_6 (μm)	1	3.35	1	3.52
W_7 (μm)	1	3.35	1	3.52
W_8 (μm)	1	3.25	2.886	9.986
W_9 (μm)	1	3.25	2.886	9.986
W_{10} (μm)	5.456	3.19	1	1.767
W_{11} (μm)	5.456	3.19	1	1.767
L (μm)	1	1	1	0.5
I_{bias} (μm)	1	1	1	0.277



(a) AC Response

(b) CMRR

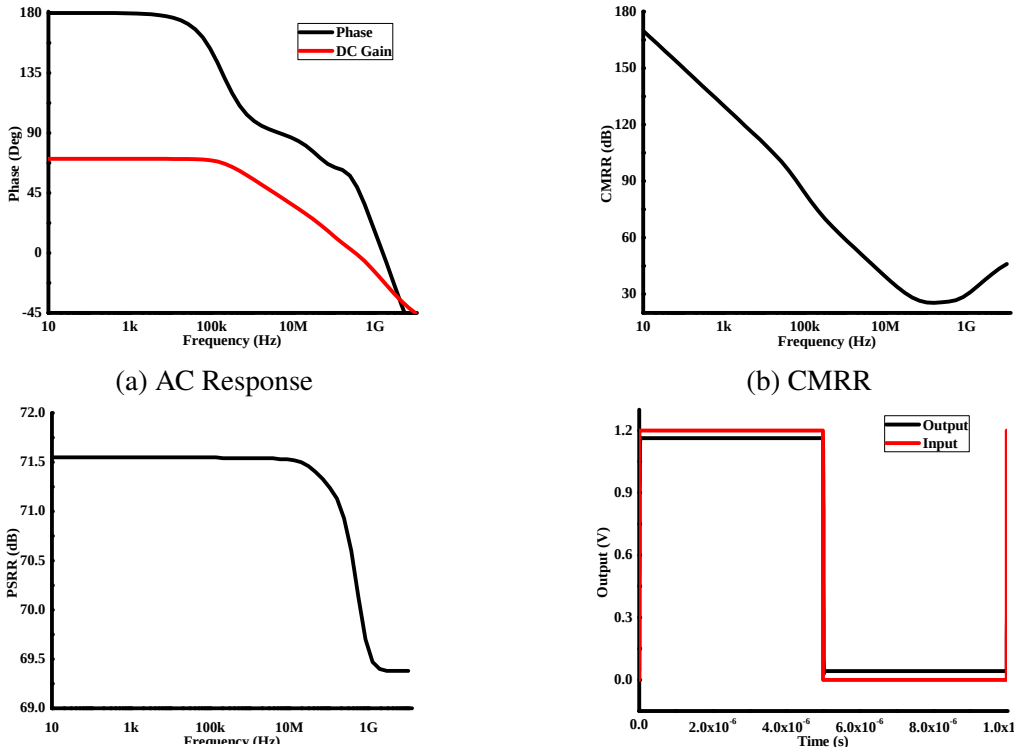
(c) PSRR

(d) Slew Rate

Figure 6.10: Simulation results for FCOTA in CMOS 180 nm technology ($V_{dd} = 1.8V$).

each iteration but the solution with the minimum overall MOS transistor area is presented in Table 6.6.

Table 6.6 demonstrates the obtained results and their comparison with competing methods. The proposed tool not only satisfies all design and performance constraints, but also minimizes the total MOS transistor area when comparison to other previous works. The simulation results demonstrates the effective performance of SCAWOA-based tool over its rivals. For example, the power consumption obtained using SCAWOA-based tool is much lower when compared to its counterparts.



(a) AC Response

(b) CMRR

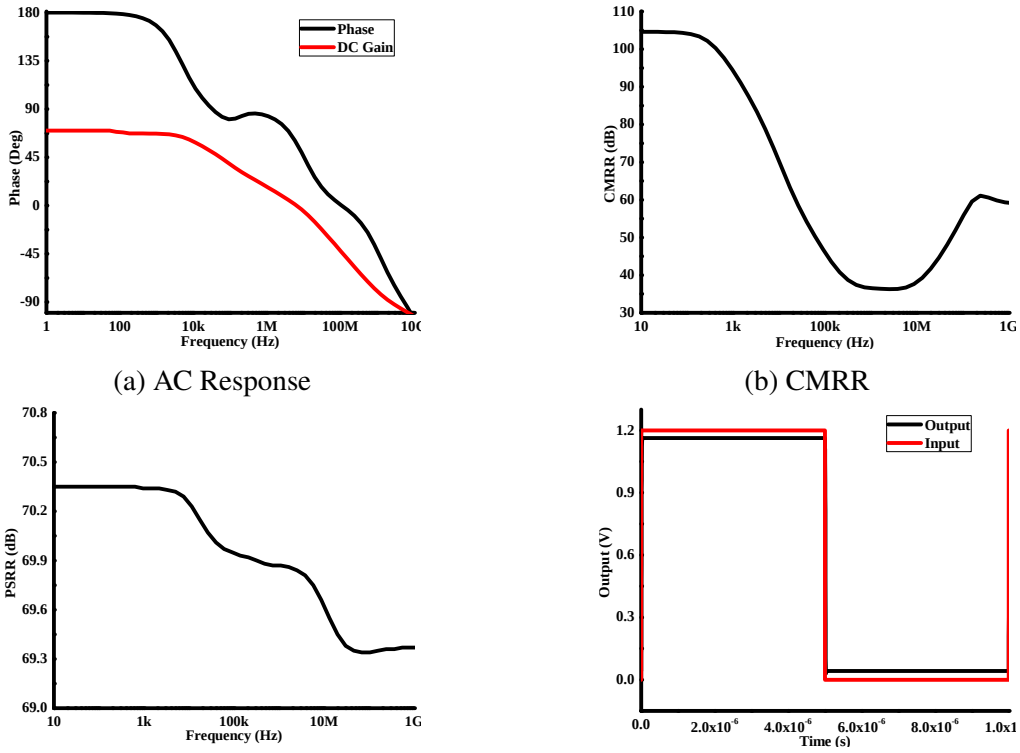
(c) PSRR

(d) Slew Rate

Figure 6.11: Simulation results for FCOTA in CMOS 130 nm technology.

6.6.3 Case study: Capacitor-less low dropout regulator

Another circuit considered for validation of the proposed simulation-based design automation tool is adaptively biased capacitor-less low-dropout regulator circuit as shown in the Figure 6.13 [135]. Conventional LDO consists of an error amplifier and the error amplifier used in this LDO is a pseudo-telescopic amplifier with load transistors (M_5 and M_6) to isolate driving transistors (M_1 and M_2) from the output through current buffer transistors (M_3 and M_4). This topology replaces the single large pass transistors in conventional topology is replaced by segmented pass transistors (M_1 and M_2) supporting low and high current loads separately. Moreover, higher load currents demand higher gain and faster operation to improve efficiency



(a) AC Response

(b) CMRR

(c) PSRR

(d) Slew Rate

Figure 6.12: Simulation results for FCOTA in CMOS 65 nm technology.

Table 6.6: Performance Summary of FCOTA.

Spec.	SCAWOA				[84]	[132]	[133]	[134]
Tech. (nm)	180	180	130	65	350	250	180	180
V_{dd} (V)	3.3	1.8	1.2	1.2	2.5	2	1.8	1.8
Vss (V)	0	0	0	0	-2.5	-2	-1.8	-1.8
Av (dB)	97.5	70.05	70.521	70.10	85.02	82.89	77 .33	84.33
UGB (MHz)	199.8	226.7	617.8	12.56	543.40	533.55	430	543.3
PM (°)	50.28	50.41	52	50.00	131.50	93.56	114	95.6
PSRR (dB)	97.76	56.1	71.5	70.35	108.20	73.23	46.5	84.37
CMRR (dB)	116	135.4	167.5	104.60	131.50	NR	196	534
SR (V/μS)	40	25	54	9.00	176.50	43	58	51.34
Pd (μW)	10.36	5.63	3.6	2.40	195	NR	6600	1200
Area (μm^2)	20.912	34.2	15.5	30.02	358.00	NR	NR	NR

when compared to lower load currents. Thus, adaptive biasing methodology is implemented using the current sensing feedback loop to meet the bias requirements of varying loads. The load capacitance of the LDO if set to be 40 pF.

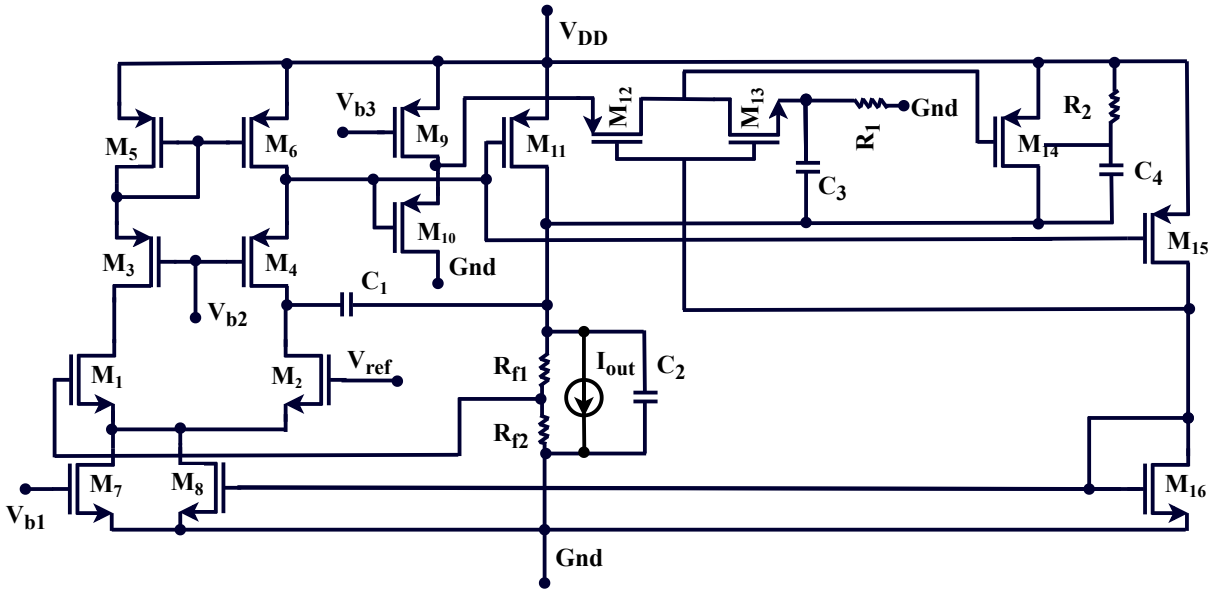


Figure 6.13: Schematic of capacitor-less low dropout regulator.

Table 6.7: Design parameters for capacitor-less low dropout regulator using SCAWOA algorithm.

Parameter	Value	Parameter	Value
W_1 (μm)	7.83	W_{15} (μm)	4.86
W_2 (μm)	7.83	W_{16} (μm)	35
W_3 (μm)	3.52	C_0 (pF)	2.5
W_4 (μm)	3.52	C_1 (pF)	40
W_5 (μm)	7.83	C_2 (pF)	5
W_6 (μm)	7.83	C_3 (pF)	5
W_7 (μm)	5	R_1 (μm)	30
W_8 (μm)	5	R_2 (μm)	5
W_9 (μm)	4.9	R_s (μm)	10
W_{10} (μm)	1.33	L (μm)	1
W_{11} (μm)	4.59	V_{b1} (V)	0.6
W_{12} (μm)	7.83	V_{b2} (V)	1.2
W_{13} (μm)	18.3	V_{b3} (V)	1.1
W_{14} (μm)	200	L_{14} (μm)	0.18

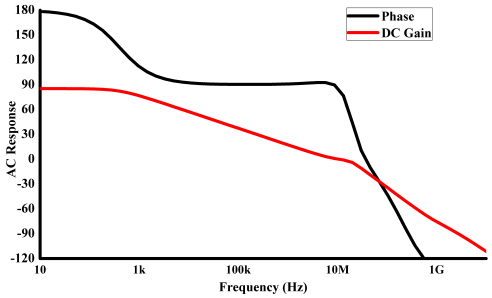
Formulation of cost function

This section discusses the process of sizing the LDO circuit shown in Figure 6.13. The inputs given to the optimization engine are design constraints and an cost function. The objective considered is optimum overshoot (OS) and undershoot (US) while satisfying all the performance constraints. The performance constraints are formulated as follows,

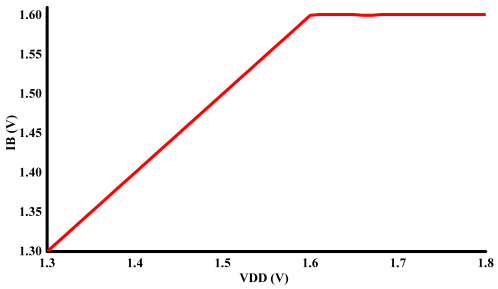
$$F(x) = V_{out+_{obt.}} - V_{out_{exp.}} + V_{out_{exp.}} - V_{out-_{obt.}} \tag{6.17}$$

$$g_i(x) = Perf._{obt.} - Perf._{exp.} \tag{6.18}$$

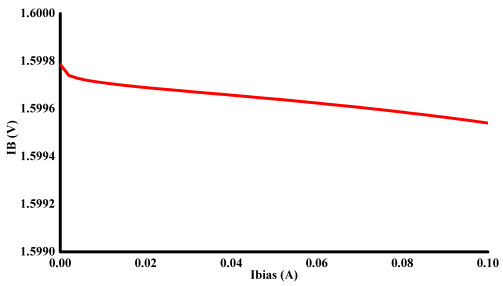
where, constraint function is the difference of the obtained value and the expected value. The



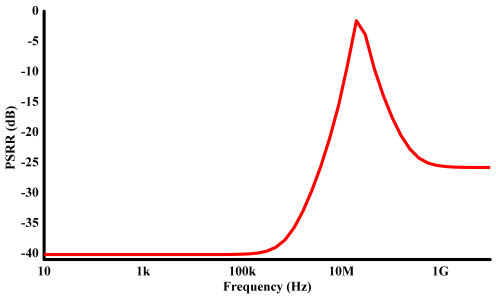
(a) AC Response



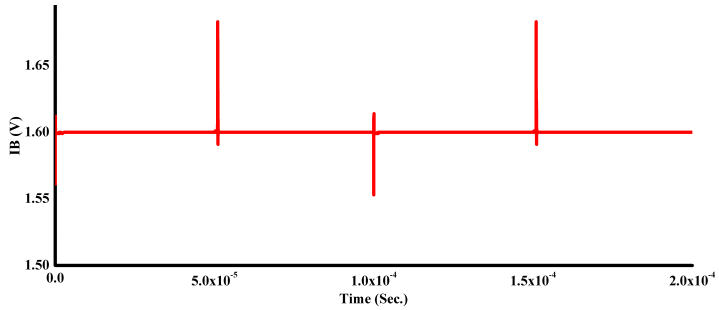
(b) Line regulation



(c) Load regulation



(d) PSRR



(e) Transient

Figure 6.14: Simulation results for LDO using simulation-based optimization technique.

obtained value is the output from the simulator and the expected value is the target specification provided by the designer. The constrained performance metrics are Loop Gain (LG), UGB,

PSRR, line regulation (LiR), load regulation (LoR) and PM. The design constraints include the range of aspect ratios and region of operation of transistors. The penalty handling technique used is static penalty function that is given in Equation 6.19.

$$P_i(x) = \sum_{i=1}^{n} (10(min(0, g_i)^2) \tag{6.19}$$

Considering the design constraints, performance constraints and cost function, the overall cost function is formulated as follows,

$$f_{obj}(x) = F(x) + w_i P_i(x) \tag{6.20}$$

The optimal design parameters of the low dropout regulator obtained by the SCAWOA-based tool are provided in Table 6.7. The design specifications of the capacitor-less LDO plotted by the proposed tool are shown in Figure 6.14a to Figure 6.14e . The proposed tool yields different feasible solutions at each iteration but the solution with the minimum objective while satisfying the performance constraints is presented in Table 6.8. The Figure of Merit is also calculated using the Equation 6.21 [136]

$$FOM_{LDO} = \frac{\Delta V_o \, I_Q}{\alpha^2 \, \Delta I_{o,max}} \tag{6.21}$$

Table 6.8: Performance summary of CMOS LDO.

Parameter	Value
V_{dd} (V)	1.8
V_o (V)	1.6
V_{ref} (V)	0.9
C_{out} (pF)	40.00
Dropout (mV)	200
Loop Gain (dB)	85.26
Phase Margin (°)	83.32
UGB (MHz)	10.91
PSRR@1MHz (dB)	36.50
PSRR@0Hz (dB)	40.23
Quiescent Current (μA)	36.00
Overshoot@100mA (mV)	83.25
Undershoot@100mA (mV)	46.83
Line Regulation (mV/mV)	0.23
Load Regulation (mV/mA)	2.25
FOM_{LDO} (μV)	15.64

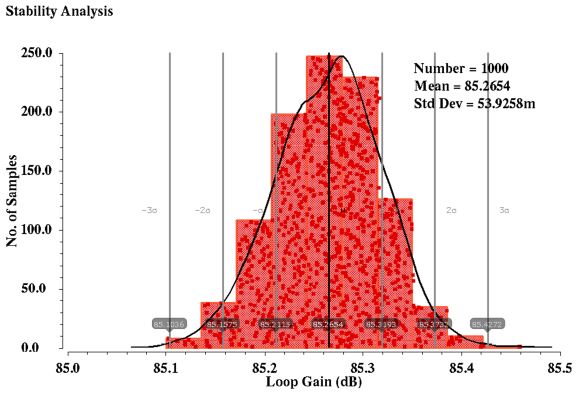
Table 6.9: Numerical results of LDO over process corners and temperature variations.

Proc. Cor.	Temp (°C)	LiR (V/mV)	LoR (V/mA)	OS (V)	US (V)	LG (dB)	PM (°)	UGB (Hz)	PSRR (dB)
FF	-25	1.81E-03	3.84E-04	5.48E-02	3.22E-02	84.00	79.40	1.29E+07	40.10
	25	2.88E-03	3.98E-04	5.47E-02	5.43E-02	72.00	76.70	1.17E+07	39.10
	125	1.54E-02	1.10E-03	7.24E-02	7.09E-02	51.00	82.90	9.08E+06	42.90
FS	-25	1.83E-03	2.29E-03	1.35E-01	3.37E-02	94.90	73.20	1.55E+07	41.70
	25	2.38E-03	2.01E-03	1.03E-01	4.62E-02	88.70	75.40	1.34E+07	39.90
	125	7.09E-03	1.07E-03	1.03E-01	7.15E-02	65.00	75.20	1.07E+07	36.90
SF	-25	1.42E-03	6.41E-04	7.20E-02	3.45E-02	89.60	91.90	9.07E+06	42.10
	25	2.14E-03	1.49E-03	6.34E-02	5.45E-02	80.70	89.60	8.81E+06	40.70
	125	8.08E-03	2.31E-03	8.17E-02	7.28E-02	56.60	91.80	7.10E+06	43.50
SS	-25	1.35E-03	1.05E-03	1.71E-01	3.81E-02	98.80	98.00	8.02E+06	43.60
	25	1.90E-03	6.65E-04	1.51E-01	5.03E-02	93.40	96.80	7.99E+06	41.90
	125	4.68E-03	8.20E-04	1.33E-01	7.91E-02	74.90	93.60	7.94E+06	38.90

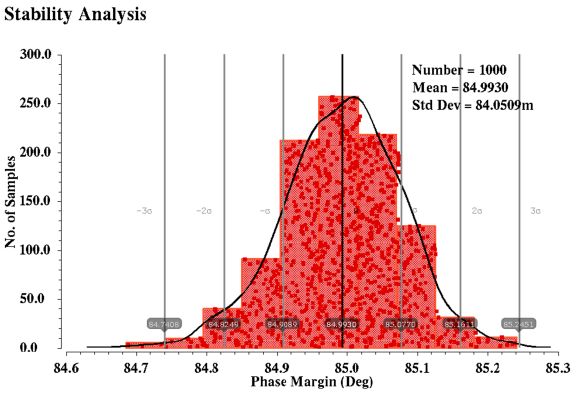
To validate the efficiency of the tool, the corner analysis is performed on the solution obtained for over 45 states as a result of cross combinations from 5 process corners, i.e., TT, SS, SF, FS and FF, supply voltage variations, i.e., V_{dd} ± 5% and temperature variations, i.e., -40 °C, 25 °C and 90 °C. Table 6.9 demonstrates the numerical results of the corner analysis for performance metrics, i.e., *LG, PM, UGB, PSRR, LiR* and *LoR*. These results satisfy the constraints imposed over the performance metrics. Furthermore, a stringent analysis for evaluating the robustness of the tool is performed using montecarlo simulation for over 1000 points and the corresponding simulation results are illustrated in Figure 6.15.

6.7 Summary

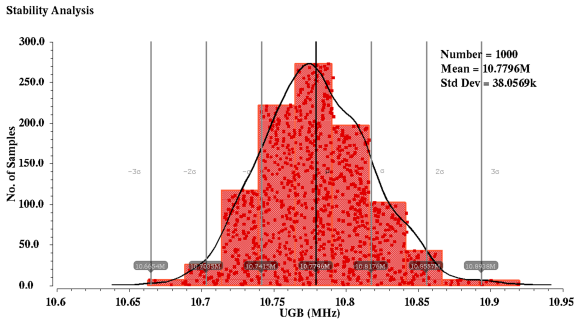
In recent years, there has been an increasing interest in the design automation of analog circuits by the researchers across the world. Various analog circuit sizing tools have been proposed in both academic research and some of them also have been appplied in industrial applications. One of the major challenges of analog sizing tools is that the optimization techniques are not powerful enough to obtain faster global optimum solution. As a step towards improving the performance of analog automation tools, a novel hybrid SCA and WOA algorithms is proposed combining the abilities of conventional algorithms. The hybrid SCAWOA has been modeled using teamwork hybrid to balance the exploration and exploitation capabilities to obtain the global optimum point. Here, the performance of SCAWOA was evaluated over a set of 23 benchmark functions and its comparison with other conventional algorithms demonstrates that the proposed algorithm re obtained solutions were more robust due to the higher ability of



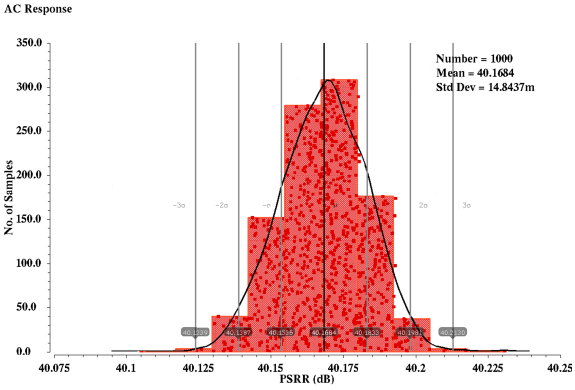
(a) Loop gain.



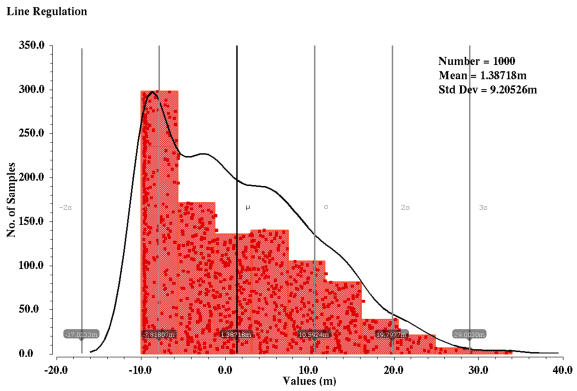
(b) Phase Margin.



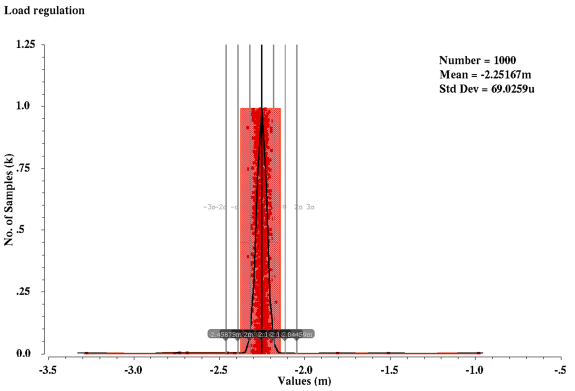
(c) Unity gain bandwidth.



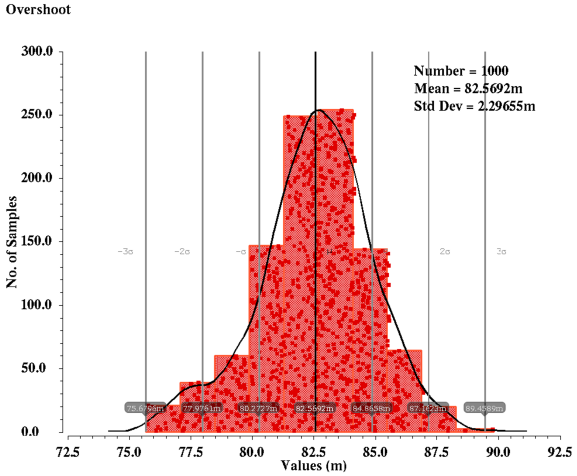
(d) PSRR



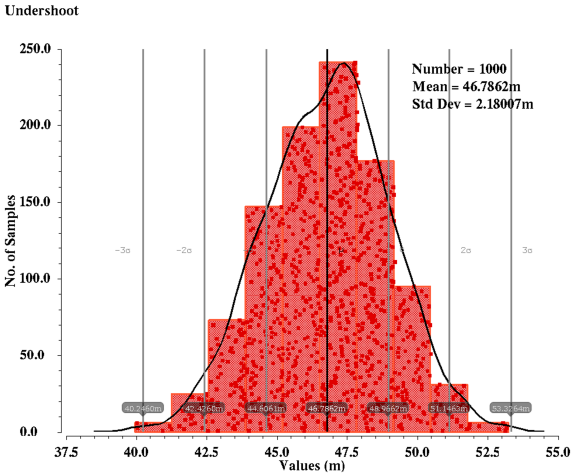
(e) Line regulation



(f) Load regulation



(g) Overshoot



(h) Undershoot

Figure 6.15: Montecarlo analysis of LDO for performance specifications over 1000 runs.

SCAWOA in the exploration and exploitation of the search space. Further, SCAWOA is used as an optimization engine of a proposed automation tool for analog IC sizing purpose.

To validate the proposed tool, a CMOS FCOTA in different technologies i.e., 180 nm (V_{DD} = 1.8V, 3.3V), 130 nm and 65 nm was considered as benchmark. The simulation results demonstrate that the tool minimizes the overall MOS transistor area and power consumption while satisfying the performance and design constraints. Also, a capacitor-less LDO is used as another benchmark for design automation with minimum undershoot and overshoot as objectives while satisfying performance constraints. The results demonstrate the overshoot and undershoot of 83.25 mV and 46.83 mV, respectively, for the load current of 100 mA and output voltage of 1.6 V at supply voltage of 1.8 V. Its robustness is also evaluated by performing corner and montecarlo analyses. The simulation results demonstrate that the performance metrics stay within tolerable limits with less standard deviation for over 1000 points.

Chapter 7

Conclusions And Future Scope

Identifying the significance of analog circuit design automation from a view point of reducing the overall time-to-market. This thesis attempts to present two different methodologies, i.e., equation-based and simulation-based, for design automation of CMOS analog circuits as a step towards aiding the analog designers for dealing with complex, cumbersome and time consuming analog circuit design. The designs are implemented in CADENCE using 180nm standard process.

The major findings of this thesis are discussed below followed by some of the perceptions for future improvements.

7.1 Major Findings

The major findings of this thesis are discussed as follows,

- The conventional optimization algorithms, i.e., GWO and WOA, are applied for design optimization of CMOS amplifier circuits, i.e., CMOS DIFFAMP and two-stage CMOS OPAMP, to obtain minimum area while satisfying the performance and design constraints using equation-based design methodology. The GWO algorithm results in minimum areas for CMOS DIFFAMP and two-stage OPAMP of 11.01 μm^2 and 28.02 μm^2, respectively, when compared to those of 19.83 μm^2 and 33.44 μm^2 obtained using WOA algorithm. (Chapter 3)

- The enhanced version of conventional GWO algorithm, i.e., EGWO algorithm, with improved exploration ability is presented. and is tested with 23 classical benchmark functions. The statistical analysis is also performed over 20 runs to determine the robustness resulting in better perfromance, for 15 out of 23 benchmark functions, over competing algorithms. The EGWO algorithm is applied for design optimization of CMOS analog amplifiers, i.e., CMOS DIFFAMP and two-stage CMOS OPAMP, with MOS transistor area as objective. The EGWO algorithm results in optimum area for CMOS DIFFAMP and two-stage CMOS OPAMP of 18.40 μm^2 and 35.56 μm^2, respectively. To evaluate the robustness of the design, a corner analysis is also performed over different corners, temperature and supply variations. The results demonstrate that the performance metrics remain in tolerable limits for most of the cases. (Chapter 4)

- A hybrid of WOA and mGWO algorithms is presented to improve the effeciency of conventional algorithms by applying teamwork hybridization process and is tested with classical 23 benchmark functions. The proposed algorithm outperforms other algorithms for over 15 out of 23 functions. It is used for design optimization of two-stage CMOS OPAMP using equation-based design methodology with MOS transistor area and power consumption as objectives. The MOS transistor area and power consumption obtained using WOA-mGWO algorithm for two-stage OPAMP are 21.51 μm^2 and 90 μW, respectively. The result over different corners demonstrate that the desired specifications are with in the tolerable limits for most of the cases. However, the challenge faced by the above equation-based methods is the requirement of fine tuning the solution manually due to the usage of simple low level equations for the evaluation of performance metrics. (Chapter 5)

- A novel design automation methodology using simulation-based design methodology is proposed for sizing of CMOS analog circuits. Eventhough the simulation time required is higher when compared to earlier methods, the accuracy is equal to the models used by the circuit simulator with the added advantage of less human intervention during the sizing process. However, the usage of present day high computation architectures help in reducing the simulation time drastically proving the effeciency of the proposed methodology. The SCAWOA algorithm combining the abilities of SCA and WOA algorithms is proposed using the same teamwork hybridization process. This algorithm is also tested with 23 benchmark functions outperforming other algorithms for 14 out of 23 benchmark functions. (Chapter 6)

- Table 7.1 gives the performance comparison of the two-stage CMOS operational amplifier using the proposed metaheuristic algorithms, i.e., EGWO, WOA-mGWO and SCA-

Table 7.1: Performance comparison of two-stage CMOS operational amplifier using the proposed metaheuristic algorithms.

Design specs.	Tar.	Chapter 1		Chapter 2	Chapter 3	Chapter 4
		GWO	WOA	EGWO	WOA-mGWO	SCA-WOA
A_v (dB)	>60	78.35	80.13	77.43	75.3	78.25
GBW (MHz)	>3	15.97	4.29	16.95	3.32	6.3
PM (degrees)	>45	61.27	62.66	64.86	60.3	59.63
SR (V/µs)	>10	16.62	13.44	10.05	18.2	10.85
Pd (mW)	<2.5	0.16	0.266	0.094	0.09	0.071
CL(pF)	>7	7	7	7	7	7
V_{ICmin} (V)	>0.3	0.38	0.45	0.4	0.4	0.5
V_{ICmax} (V)	<1.6	1.22	1.35	1.2	1.2	1.2
CMRR (dB)	>60	92.76	92.12	86.31	83.01	79.93
PSRR+ (dB)	>70	78.27	80.07	87.5	84.3	85.12
PS RR- (dB)	>70	72.25	75.54	78.65	77.45	81.61
Area (µm^2)	Obj.	28.02	33.44	35.56	21.51	19.31
FOM_1	Max.	8.6	2.31	6.818	1.225	2.205
FOM_2	Max.	24.93	3.38	35.49	12.004	32.17
Tech. (µm)		0.18	0.18	0.18	0.18	0.18

WOA, in chapters 4, 5 and 6, respectively. The SCAWOA algorithm converges to give lower area and optmum power consumption when compared to other algorithms. The proposed algorithms help designers in achieving low power and compact size resulting in reduced cost and overall time-to-market.

- Considering the efficiency of SCAWOA algorithm over other competing algorithm for its application to analog sizing problems, it is applied in the optimization engine of the simulation-based methodology for circuit sizing of complex CMOS analog circuits, i.e., CMOS FCOTA and capacitor-less LDO circuits, with comparitively more number of transistors. The FCOTA is designed for different CMOS technologies, i.e., 180nm (3.3 V and 1.8 V), 130nm and 65nm, with the objective of obtaining the optimum MOS transistor area while satisfying performance and design constraints. The tool demonstrates better performance when compared to state-of-the-art designs. The capacitor-less LDO is also implemented in CMOS standard process with the objective of reducing overall undershoot and overshoot. The obtained results demonstrate the overshoot and undershoot of 83.25 mV and 46.83 mV, respectively, for the load current of 100 mA and output voltage of 1.6 V at supply voltage of 1.8 V. To test the robustness of the design, PVT and montecarlo analyses were performed. The results demonstrate that the performance metrics, i.e., LG, PM, UGB, PSRR, LiR, LoR, US and OS, for over 1000 points stay within tolerable limits with less standard deviation. (Chapter 6)

7.2 Future Work

Some potential extensions of the work presented in this thesis are:

- According to NFL theorem, there exists no optimziation algorithm that can solve all optimization problems. Thus, there is a requirement of novel robust optimization algorithms or improved versions of conventional optimization algorithms for solving various global optimization problems.
- High-level design equations can be used in the equation-based circuit sizing methodology for improving the accuracy and effeciency. This avoids the fine tuning of the parameters manually.
- Metaheuristic algorithms can be applied at various levels in automation of layout design, i.e., floorplanning, placement and routing.
- Instead of converting multiple objectives into a single objective, multi-objective optimization algorithms can be employed to generate the pareto fronts. This helps the designer to choose from various solutions and corresponding design parameters based on the requirement.

Appendix A

A.1 Analog Circuit Design [117]

The empirical interpretation for the design of CMOS differential amplifier and two-stage miller compensated operational amplifier are as follows:

A.1.1 CMOS differential amplifier

Step 1: The range of Drain current of M_5 (I_{D5}) is obtained to satisfy slew-rate (SR).

$$SR = \frac{I_{D5}}{C_L} \tag{A.1}$$

$$f_{-3dB} = \frac{1}{R_{out} \cdot C_L} \tag{A.2}$$

Step 2:Aspect Ratio of Transistor $M_1(M_2), i.e., S_1(= S_2)$, is determined in order to satisfy DC gain (A_V), where

$$A_V = \frac{\sqrt{4K_n' S_1}}{\left(\lambda_n + \lambda_p\right)\sqrt{I_{D5}}} \tag{A.3}$$

Step 3: Determine $S_3(= S_4)$ to satisfy the positive Input Common Mode Range (V_{ICmax})

$$V_{ICmax} = V_{DD} - V_{SG3} + V_{tn1} \tag{A.4}$$

$$S_3 = \frac{2I_{D5}}{K_p'(V_{SG3} + V_{tp})^2} \tag{A.5}$$

Step 4: The value of S_5 is determined to satisfy negative Input Common Mode Range V_{ICmin}

$$V_{ICmin} = V_{SS} - V_{DS5sat} + V_{SG1} \tag{A.6}$$

$$S_5 = \frac{2 \cdot I_{D5}}{K_n' \left(V_{D5sat}\right)^2} \tag{A.7}$$

Step 5: The I_{D5} is determined to satisfy power dissipation (P_d)

$$P_d = I_{D5} \left(V_{DD} + |V_{SS}|\right) \tag{A.8}$$

A.1.2 Two-stage miller compensated CMOS operational amplifier

Step 1: A small value of C_C is chosen to place the second pole about 2.2 times greater than the UGB and to achieve the phase margin of 60 ° , The right hand plane zero is assumed to be 10 times beyond UGB.

$$C_C > 0.22 \cdot C_L \tag{A.9}$$

$$P_2 = -\frac{g_{m6}}{C_L} \tag{A.10}$$

$$z_1 = \frac{g_{m6}}{C_C} \tag{A.11}$$

Step 2: I_{D5} is obtained to meet the desired specification of slew rate.

$$I_{D5} = SR \cdot C_C \tag{A.12}$$

Step 3: The input transconductance of transistors M_1 and M_2 is determined from UGB and C_C.

$$g_{m1} = 2\pi \cdot UGB \cdot C_C \tag{A.13}$$

Step 4: Determine $S_1 (= S_2)$ using following equation

$$S_1 = \frac{g_{m1}}{K_n' \cdot I_{D5}} \tag{A.14}$$

Step 5: Maximum value of ICMR is used to determine $S_3 (= S_4)$

$$S_3 = \frac{I_{D5}}{K_p' \left(V_{DD} - V_{inmax} - \left|V_{tpmax}\right| + V_{tnmin}\right)^2} \tag{A.15}$$

Step 6: Minimum value of ICMR is used to determine $S_5 (= S_8)$

$$S_5 = \frac{2 \cdot I_{D5}}{K_n' \left(V_{D5sat}\right)^2} \tag{A.16}$$

where $V_{D5sat} = V_{inmin} - V_{SS} - V_{inmax} - \sqrt{\frac{I_{D5}}{K_n' \cdot S_1}}$

Step 7: To estimate S_6, we have

$$S_6 = \frac{S_4 \cdot g_{m6}}{g_{m4}} \tag{A.17}$$

where $g_{m4} = \sqrt{K_p'} \cdot S_4 \cdot I_{D5}$

Step 8: The current I_{D6} is required for power dissipation (P_d).

$$I_{D6} = \frac{(g_{m6})^2}{2K_p' \cdot S_6} \tag{A.18}$$

Step 9: In order to attain the current ratio between I_{D5} and I_{D6}, we evaluate the value of S_7 as follows,

$$S_7 = \frac{S_5 \cdot I_{D6}}{I_{D5}} \tag{A.19}$$

Step 10: The values of gain (A_V) and power dissipation (P_d) are estimated using following equations:

$$A_V = \frac{2g_{m2} \cdot g_{m6}}{I_{D6} \cdot I_{D5} \cdot \left(\lambda_n + \lambda_p\right)^2} \tag{A.20}$$

$$P_d = (I_{D5} + I_{D6}) \cdot (V_{DD} + |V_{SS}|) \tag{A.21}$$

A.2 Benchmark Functions

The performance of the algorithm is compared with other algorithms with respect to 23 classical and popular benchmark functions (Nenavath and Kumar Jatoth, 2017), as shown in Table A.1, to analyze the efficiency of the proposed algorithm. These benchmark functions are classified as unimodal ($F_1 - F_7$), fixed low dimensional($F_8 - F_{13}$) and multimodal high dimensional ($F_{14} - F_{23}$) benchmark functions. The statistical analysis is performed over these benchmark function to obtain mean, median, best, worst and standard deviation.

Table A.1: Benchmark functions used for the experimental study (D: Dimension, F_{min}: Global minima).

Function	Formulation	D	Range	f_{min}
Sphere	$f_1(x)=\sum_{i=1}^{D} x_i^2$	30	[-100, 100]	0
Schwefel 2.22	$f_2(x)=\sum_{i=1}^{D}\lvert x_i\rvert+\prod_{i=1}^{D}\lvert x_i\rvert$	30	[-10, 10]	0
Schwefel 1.2	$f_3(x)=\sum_{i=1}^{D}\left(\sum_{j=1}^{i} x_j^2\right)$	30	[-100, 100]	0
Schwefel 2.21	$f_4(x)=\max_i\{\lvert x_i\rvert, 1\le i\le D\}$	30	[-100, 100]	0
Rosenbrock	$f_5(x)=\sum_{i=1}^{D-1}\left[100\left(x_{i+1}-x_i^2\right)^2+\left(x_i-1\right)^2\right]$	30	[-30, 30]	0
Step	$f_6(x)=\sum_{i=1}^{D}\left(\lfloor x_i+0.5\rfloor\right)^2$	30	[-100, 100]	0
Quartic	$f_7(x)=\sum_{i=1}^{D} i x_i^4+random[0,1)$	30	[-1.28, 1.28]	0
Schwefel	$f_8(x)=-\sum_{i=1}^{D}\left(x_i\sin\sqrt{\lvert x_i\rvert}\right)$	30	[-500, 500]	-418.9829*D
Rastrigin	$f_9(x)=10D+\sum_{i=1}^{D}\left(x_i^2-10\cos(2\pi x_i)\right)$	30	[-5.12, 5.12]	0
Ackley	$f_{10}(x)=-20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right)-\exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right)+20+e$	30	[-32, 32]	0
Griewank	$f_{11}(x)=\frac{1}{4000}\sum_{i=1}^{D} x_i^2-\prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	30	[-600, 600]	0
Penalized	$f_{12}(x)=\frac{\pi}{D}\left\{10\sin(\pi y_1)+\sum_{i=1}^{D-1}(y_i-1)^2\left[1+10\sin^2(\pi y_{i+1})\right]+(y_D-1)^2\right\}+\sum_{i=1}^{D} u(x_i,10,100,4)$ where $y_i=1+\frac{x_i+1}{4}$, and $u(x_i,a,k,m)=\begin{cases}k(x_i-a)^m; x_i>a\\ 0; -a<x_i<a\\ k(-x_i-a)^m; x_i<-a\end{cases}$	30	[-50, 50]	0

Penalize 2	$f_{13}(x)=0.1\left\{\sin^2(3\pi x_1)+\sum_{i=1}^{D}(x_i-1)^2\left[1+\sin^2(3\pi x_i)+1\right]+(x_n-1)^2\left[1+\sin^2(2\pi x_n)\right]\right\}+\sum_{i=1}^{D}u(x_i,5,100,4)$	30	[-50, 50]	0
Foxholes	$f_{14}(x)=\left[\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{D}(x_i-a_{ij})^6}\right]^{-1}$	2	[-65.536, 65.536]	0.998004
Kowalik	$f_{15}(x)=\sum_{i=1}^{11}\left[a_i-\frac{x_1(b_i^2+b_i x_2)}{b_i^2+b_i x_3+x_4}\right]^2$	4	[-5, 5]	0.000307
Six-hump Camel-Back	$f_{16}(x)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$	2	[-5, 5]	-1.03162
Branin	$f_{17}(x)=\left(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6\right)^2+10\left(1-\frac{1}{8\pi}\right)\cos x_1+10$	2	[-5,5]	0.398
Goldstein-Price	$f_{18}(x)=\left[1+(x_1+x_2+1)^2(10-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)\right]*\left[30+(2x_1-3x_2^2)\right.$ $\left.(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2)\right]$	2	[-5,5]	3
Hartman 3	$f_{19}(x)=-\sum_{i=1}^{4}c_i\exp\left[-\sum_{j=1}^{3}a_{ij}(x_j-p_{ij})^2\right]$	3	[-5,5]	-3.86278
Hartman 6	$f_{20}(x)=-\sum_{i=1}^{4}c_i\exp\left[-\sum_{j=1}^{6}a_{ij}(x_j-p_{ij})^2\right]$	6	[-5, 5]	-3.32236
Shekel5	$f_{21}(x)=-\sum_{i=1}^{5}\left[(x-a_i)(x-a_i)^T+c_i\right]^{-1}$	3	[-5, 5]	-10.1532
Shekel7	$f_{22}(x)=-\sum_{i=1}^{7}\left[(x-a_i)(x-a_i)^T+c_i\right]^{-1}$	3	[-5, 5]	-10.4029
Shekel10	$f_{23}(x)=-\sum_{i=1}^{10}\left[(x-a_i)(x-a_i)^T+c_i\right]^{-1}$	3	[-5, 5]	-10.5364

A.3 Comparison of LRH and LTH models

Table A.2: Minimization results of 23 benchmark functions for WOA-mGWO.

Function		WOA-mGWO (LRH)	WOA-mGWO (LTH)
F1	Mean	0	0
	Best	0	0
	Worst	0	0
	SD	0	0
F2	Mean	1.05E-258	0
	Best	1.02E-269	0
	Worst	2.11E-257	0
	SD	0	0
F3	Mean	2.34E-61	1.23E-87
	Best	2.92E-81	5.30E-142
	Worst	4.68E-60	1.67E-86
	SD	1.05E-60	2.75E-85
F4	Mean	5.98E-92	9.30E-196
	Best	3.22E-105	4.30E-207
	Worst	1.19E-90	1.80E-194
	SD	2.66E-91	0
F5	Mean	26.0449	24.3451
	Best	25.039	23.8722
	Worst	28.7357	26.9609
	SD	0.8882	0.266
F6	Mean	0.1768	7.43E-07
	Best	0.0001	2.88E-07
	Worst	0.6185	1.18E-06
	SD	0.2138	5.59E-02
F7	Mean	0.0001	6.71E-05
	Best	8.13E-05	3.08E-06
	Worst	0.0003	1.45E-04
	SD	7.94E-05	4.21E-05
F8	Mean	-12237.6933	-12011.81
	Best	-12569.4616	-12569.48
	Worst	-11032.4669	-8739.81
	SD	481.6351	820.85

Function		WOA-mGWO (LRH)	WOA-mGWO (LTH)
F9	Mean	0	0
	Best	0	0
	Worst	0	0
	SD	0	0
F10	Mean	3.55E-15	3.20E-15
	Best	8.88E-16	8.88E-16
	Worst	4.44E-15	4.44E-15
	SD	1.58E-15	1.67E-15
F11	Mean	0	0
	Best	0	0
	Worst	0	0
	SD	0	0
F12	Mean	0.020468592	0.0016
	Best	7.99E-05	7.40E-08
	Worst	0.1018	0.0065
	SD	0.0232	0.0023
F13	Mean	0.5792	0.0235
	Best	0.0083	2.13E-06
	Worst	1.0826	0.2479
	SD	0.3211	0.0538
F14	Mean	1.2452	0.998
	Best	0.998	0.998
	Worst	2.5444	0.998
	SD	0.7445	2.24E-14
F15	Mean	0.0004	3.19E-04
	Best	0.0003	3.07E-04
	Worst	0.001	5.16E-04
	SD	0.0002	2.81E-04
F16	Mean	-1.0316	-1.0316
	Best	-1.0316	-1.0316
	Worst	-1.0316	-1.0316
	SD	1.26E-16	1.28E-16
F17	Mean	0.3978	0.3978
	Best	0.3978	0.3978
	Worst	0.3978	0.3978
	SD	0	1.73E-10

Function		WOA-mGWO (LRH)	WOA-mGWO (LTH)
F18	Mean	3	3
	Best	3	3
	Worst	3	3
	SD	4.33E-16	1.39E-06
F19	Mean	-3.8618	-3.8627
	Best	-3.8627	-3.8627
	Worst	-3.8588	-3.8626
	SD	0.0012	5.58E-05
F20	Mean	-3.3002	-3.2979
	Best	-3.3219	-3.3219
	Worst	-3.2037	-3.2006
	SD	0.0534	5.80E-02
F21	Mean	-8.6237	-9.8983
	Best	-10.1531	-10.1532
	Worst	-5.0551	-5.05519
	SD	2.3968	1.63E-07
F22	Mean	-7.9428	-10.4029
	Best	-10.4029	-10.4029
	Worst	-3.7242	-10.4029
	SD	2.69E-05	1.58E-06
F23	Mean	-7.2916	-9.9956
	Best	-10.5364	-10.5364
	Worst	-5.1284	-5.1284
	SD	2.7181	4.28E-07

Table A.3: Minimization results of 23 benchmark functions for SCA-WOA.

Function		SCA-WOA (LRH)	SCA-WOA (LTH)
F1	Mean	4.56E-179	3.66E-193
	Best	8.6E-192	2.07E-201
	Worst	9.11E-178	6.18E-192
	SD	0	0
F2	Mean	5.54E-108	5.66E-111
	Best	4.4E-118	1.54E-123
	Worst	4.47E-107	1.12E-109
	SD	1.19E-107	2.49E-110
F3	Mean	2.10E-67	1.03E-50
	Best	8.81E-105	1.54E-71
	Worst	4.19E-66	2.02E-49
	SD	9.38E-67	4.52E-50
F4	Mean	7.01E-59	8.96E-77
	Best	1.97E-74	5.52E-88
	Worst	1.40E-57	1.63E-65
	SD	3.14E-58	3.63E-76
F5	Mean	24.8276	5.9676
	Best	24.2841	5.3359
	Worst	26.0154	6.2331
	SD	0.4417	0.2198
F6	Mean	5.83E-05	0.0000114
	Best	4.55E-06	0.00000356
	Worst	3.62E-05	0.0000198
	SD	7.31E-06	0.00000598
F7	Mean	6.41E-03	0.0001
	Best	2.72E-06	0.000000917
	Worst	0.0005	0.0004
	SD	4.24E-04	0.000096
F8	Mean	-12004.22	-3235.6
	Best	-12569.48	-4189.8
	Worst	-8399.26	-2348.1
	SD	1272.39	646.04

Function		SCA-WOA (LRH)	SCA-WOA (LTH)
F9	Mean	0	0
	Best	0	0
	Worst	0	0
	SD	0	0
F10	Mean	3.73E-15	3.16E-15
	Best	8.88E-16	8.88E-16
	Worst	4.44E-15	4.44E-15
	SD	1.76E-15	7.94E-16
F11	Mean	0	0
	Best	0	0
	Worst	0	0
	SD	0	0
F12	Mean	0.0051	0.0045
	Best	6.90E-06	0.0000059
	Worst	0.0245	0.0197
	SD	0.0082	0.0074
F13	Mean	0.0782	0.0375
	Best	1.67E-04	0.0000175
	Worst	0.3646	0.1971
	SD	0.1064	0.0545
F14	Mean	0.998	0.998
	Best	0.998	0.998
	Worst	0.998	0.998
	SD	2.23E-14	0
F15	Mean	0.0004	0.0004
	Best	0.0003	0.0003
	Worst	0.001	0.0006
	SD	0.0002	0.0000553
F16	Mean	-1.0316	-1.0316
	Best	-1.0316	-1.0316
	Worst	-1.0316	-1.0316
	SD	2.27E-16	1.36E-16
F17	Mean	0.3978	0.3978
	Best	0.3978	0.3978
	Worst	0.3978	0.3978
	SD	0	0

Function		SCA-WOA (LRH)	SCA-WOA (LTH)
F18	Mean	3	3
	Best	3	3
	Worst	3	3
	SD	7.22E-16	3.47E-16
F19	Mean	-3.8626	-3.8625
	Best	-3.8627	-3.8627
	Worst	-3.8621	-3.8618
	SD	0.0001	0.0002
F20	Mean	-3.2795	-3.322
	Best	-3.3219	-3.322
	Worst	-3.2031	-3.322
	SD	0.0593	0.00000207
F21	Mean	-9.8982	-9.3878
	Best	-10.1531	-10.1531
	Worst	-5.0551	-5.0551
	SD	1.9399	1.8673
F22	Mean	-10.4029	-10.4029
	Best	-10.4029	-10.4029
	Worst	-10.4029	-10.4029
	SD	3.09E-06	3.35E-12
F23	Mean	-9.4548	-9.9946
	Best	-10.5364	-10.5364
	Worst	-5.1284	-5.1285
	SD	2.2193	1.6642

A.4 Performance Comparison

Table A.4: Relative performance comparison of metaheuristic algorithms for CMOS differential amplifier.

Design Criteria	Specs.	GSA	GWO	PSO	WOA	EGWO	WOA-mGWO	SCA WOA
SR (V/µs)	>10	19.59	39.74	17.92	14.79	11.57	10.38	11.85
$P_d(\mu W)$	<200	71.62	73.06	92.45	70.2	60.75	54.91	35.91
PM (°)	>45	89.44	87.75	87.27	89.54	85.4	86.5	87.26
$f_{-3dB}(KHz)$	>200	91.2	457	155.9	154	277.8	207.93	206.3
$A_v(dB)$	>40	45.07	45.8	44.06	46.06	43.06	46.84	44.8
CMRR (dB)	>60	82.66	87.35	78.5	85.49	85.05	59.84	82.17
PSRR (dB)	>70	90.09	78.46	87.95	92.12	90.12	69.55	75.8
MOS Area (μm^2)	Obj.	22.12	11.01	25.57	19.83	18.4	15.38	12.17
Technology		0.18	0.18	0.18	0.18	0.18	0.18	0.18

Table A.5: Relative performance comparison of metaheuristic algorithms for two-stage CMOS operational amplifier.

Design Criteria	Specs.	GSA	GWO	PSO	WOA	EGWO	WOA-mGWO	SCA WOA
$A_v(dB)$	>60	72.74	78.35	66.8	80.13	77.43	75.3	78.25
GBW (MHz)	>3	10	15.97	11.31	4.29	16.95	3.32	6.3
PM (°)	>45	81.4	61.27	61	62.66	64.86	60.3	59.63
SR (V/µs)	>10	11.2	16.62	14.3	13.44	10.05	18.2	10.85
$P_d(mW)$	<2.5	0.13	0.16	0.17	0.266	0.094	0.09	0.071
$C_L(pF)$	>7	7	7	7	7	7	7	7
$V_{ICmin\}(V)}$	>0.3	0.8	0.38	0.75	0.45	0.4	0.4	0.5
$V_{ICmax\}(V)}$	<1.6	1.6	1.22	1.39	1.35	1.2	1.2	1.2
CMRR (dB)	>60	84.64	92.76	85.91	92.12	86.31	83.01	79.93
PSRR+ (dB)	>70	88.14	78.27	67.3	80.07	87.5	84.3	85.12
PSRR- (dB)	>70	77.24	72.25	91.17	75.54	78.65	77.45	81.61
MOS Area (μm^2)	Obj.	35.08	28.02	47.6	33.44	35.56	21.51	19.31
Technology		0.18	0.18	0.18	0.18	0.18	0.18	0.18

A.5 Sample scripts

A.5.1 Source code for WOA-mGWO algorithm

The sample source code for the proposed WOA-mGWO algorithm is presented below for proper understanding of the optimization process.

```
SearchAgents_no :  Number of search agents
Max_iter :  Maximum number of iterations (termination criteria)
lb&ub :  Lower and upper bounds, respectively
dim :  Dimension
fobj :  Objective/Cost function


% Hybrid Whale Optimization Algorithm and modified Grey Wolf Optimization
Algorithm
function[Alpha_score,Alpha_pos,Convergence_curve]=WOAGWO2(SearchAgents_no,Max_iter,
% initialize alpha, beta, and delta_pos (Position of alpha, beta and
delta wolves)
Alpha_pos=zeros(1,dim);
Alpha_score=inf; % inf for minimization problems and -inf for
maximization problems
Beta_pos=zeros(1,dim);
Beta_score=inf; % inf for minimization problems and -inf for maximization
problems
Delta_pos=zeros(1,dim);
Delta_score=inf; % inf for minimization problems and -inf for
maximization problems
%Initialize the positions of search agents
Positions=initialization(SearchAgents_no,dim,ub,lb);
Convergence_curve=zeros(1,Max_iter);
l=0; % Loop counter
% Main loop
while l<Max_iter
for i=1:size(Positions,1)
```

```
% Return back the search agents that go beyond the boundaries of the
search space
Flag4ub=Positions(i,:)>ub;
Flag4lb=Positions(i,:)<lb;
Positions(i,:)=(Positions(i,:).*( (Flag4ub+Flag4lb)))+ub.*Flag4ub+lb.*Flag4lb;
% Calculate objective function for each search agent
Fitness=fobj(Positions(i,:));
% Update Alpha, Beta, and Delta
if fitness<Alpha_score
Alpha_score=fitness; % Update alpha
Alpha_pos=Positions(i,:);
end
if fitness>Alpha_score&& fitness<Beta_score
Beta_score=fitness; % Update beta
Beta_pos=Positions(i,:);
end
if fitness>Alpha_score&& fitness>Beta_score&& fitness<Delta_score
Delta_score=fitness; % Update delta
Delta_pos=Positions(i,:);
end
end
a = 2 - (l^2) * ((2)/(Max_iter^2)); % 'a' decreases from 2 to 0 Equation 5.5
a2 = -1 + (l^2) * ((-1)/(Max_iter^2));
% Update the Position of search agents including omegas
for i=1:size(Positions,1)
r1=rand(); % 'r1' is a random number in [0,1]
r2=rand(); % 'r2' is a random number in [0,1]
b = 1;
A1=2*a*r1-a; % Equation (5.3)
C1=2*r2; % Equation (5.4)
l1= (a2-1)*rand+1; % random number of range [-1 1]
p = rand();
for j=1:size(Positions,2)
if p<0.5
if A1<1
% Position update of search agent 'i' with jth dimension considering
```

```
three best solutions (mGWO).
D_Alpha=abs(C1.*Alpha_pos(j)-Positions(i,j)); % Equation (5.6a)
X1=Alpha_pos(j)-A1.*D_Alpha; % Equation (5.7a)
D_Beta=abs(C1*Beta_pos(j)-Positions(i,j)); % Equation (5.6b)
X2=Beta_pos(j)-A1.*D_Beta; % Equation (5.7b)
D_Delta=abs(C1.*Delta_pos(j)-Positions(i,j)); % Equation (5.6c)
X3=Delta_pos(j)-A1.*D_Delta; % Equation (5.7c)
Positions(i,j)=(X1+X2+X3)/3; % Equation (5.8)
elseif A1>=1
% Position update of the search agent using the random search agent from
the population, instead of best search agents.
rand_leader_index = floor(SearchAgents_no*rand()+1);
X_rand = Positions(rand_leader_index, :);
D_X_rand=abs(C1.*X_rand(j)-Positions(i,j)); % Equation (5.9)
Positions(i,j)=X_rand(j)-A1.*D_X_rand; % Equation (5.10)
end
elseif p>=0.5
distance2Alpha=abs(Alpha_pos(j)-Positions(i,j)); % Equation (5.11a)
X1=distance2Alpha*exp(b.*l1).*cos(l.*2*pi)+Alpha_pos(j); % Equation
(5.12a)
distance2Beta=abs(Beta_pos(j)-Positions(i,j)); % Equation (5.11b)
X2=distance2Beta*exp(b.*l1).*cos(l.*2*pi)+Beta_pos(j); % Equation (5.12b)
distance2Delta=abs(Delta_pos(j)-Positions(i,j)); % Equation (5.11c)
X3=distance2Delta*exp(b.*l1).*cos(l.*2*pi)+Delta_pos(j); % Equation
(5.12c)
Positions(i,j)=(X1+X2+X3)/3; % Equation (5.13)
end
end
end
l=l+1;
Convergence_curve(l)=Alpha_score;
end
```

A.5.2 Sample OCEAN script

The sample OCEAN script that is used for design automation of CMOS analog circuits while interfacing MATLAB and CADENCE.

```
simulator( 'spectre )
design("/home/CDC002/simulation/opamp/spectre/schematic/netlist")
resultsDir( "/home/CDC002/simulation/opamp/spectre/schematic" )
modelFile('("/home/CDC002/Designto/../Models/Spectre/l65ll_v181.lib.scs"
"tt_ll_rvt12")
stimulusFile( ?xlate nil "../spectre/schematic/netlist/_stimuli.scs")
analysis('dc ?saveOppoint t )
analysis('ac ?start "10" ?stop "10G" )
q = infile( "/home/CDC002/Designto/circuits/opamp/designparam.txt" )
fscanf( q "%s %s %s %s %s %s %s" w12 w34 w58 w6 w7 cl ib )
desVar("cl" cl )
desVar("w1" w12 )
desVar("w2" w12 )
desVar("w3" w34 )
desVar("w4" w34 )
desVar("w5" w58 )
desVar("w6" w6 )
desVar("w7" w7 )
desVar("w8" w58 )
desVar("ib" ib )
pp = infile( "/home/CDC002/Designto/circuits/opamp/rcx.txt" )
fscanf( pp "%s %s %s" cc r vi ) desVar("cc" cc )
desVar("r" r )
desVar("vi" vi )
envOption( 'analysisOrder list("dc" "ac") )
temp( 27 )
run()
ocnPrint(?output "spec.txt" pv("M1" "ids" ?result "dcOpInfo") pv("M2"
"ids" ?result "dcOpInfo") pv("M3" "ids" ?result "dcOpInfo") pv("M4"
"ids" ?result "dcOpInfo") pv("M5" "ids" ?result "dcOpInfo") pv("M6"
"ids" ?result "dcOpInfo") pv("M7" "ids" ?result "dcOpInfo") pv("M8" "ids"
```

```
?result "dcOpInfo") ?numberNotation 'scientific)
ocnPrint(?output " spec.txt " pv("M1" "gm" ?result "dcOpInfo") pv("M2"
"gm" ?result "dcOpInfo") pv("M3" "gm" ?result "dcOpInfo") pv("M4"
"gm" ?result "dcOpInfo") pv("M5" "gm" ?result "dcOpInfo") pv("M6" "gm"
?result "dcOpInfo") pv("M7" "gm" ?result "dcOpInfo") pv("M8" "gm" ?result
"dcOpInfo") ?numberNotation 'scientific)
ocnPrint(?output " spec.txt " value(dB20((VF("/out") / VF("/vin2")))
20) phaseMargin((VF("/out") / VF("/vin2"))) gainBwProd((VF("/out") /
VF("/vin2"))) ?numberNotation 'scientific)
ocnPrint(?output " spec.txt " pv("M1" "vds" ?result "dcOpInfo") pv("M2"
"vds" ?result "dcOpInfo") pv("M3" "vds" ?result "dcOpInfo") pv("M4"
"vds" ?result "dcOpInfo") pv("M5" "vds" ?result "dcOpInfo") pv("M6"
"vds" ?result "dcOpInfo") pv("M7" "vds" ?result "dcOpInfo") pv("M8" "vds"
?result "dcOpInfo") ?numberNotation 'scientific)
ocnPrint(?output " spec.txt " pv("M1" "vdsat" ?result "dcOpInfo") pv("M2"
"vdsat" ?result "dcOpInfo") pv("M3" "vdsat" ?result "dcOpInfo") pv("M4"
"vdsat" ?result "dcOpInfo") pv("M5" "vdsat" ?result "dcOpInfo") pv("M6"
"vdsat" ?result "dcOpInfo") pv("M7" "vdsat" ?result "dcOpInfo") pv("M8"
"vdsat" ?result "dcOpInfo") ?numberNotation 'scientific)
ocnPrint(?output " spec.txt " pv("M1" "vgs" ?result "dcOpInfo") pv("M2"
"vgs" ?result "dcOpInfo") pv("M3" "vgs" ?result "dcOpInfo") pv("M4"
"vgs" ?result "dcOpInfo") pv("M5" "vgs" ?result "dcOpInfo") pv("M6"
"vgs" ?result "dcOpInfo") pv("M7" "vgs" ?result "dcOpInfo") pv("M8" "vgs"
?result "dcOpInfo") ?numberNotation 'scientific)
ocnPrint(?output " spec.txt " pv("M1" "vth" ?result "dcOpInfo") pv("M2"
"vth" ?result "dcOpInfo") pv("M3" "vth" ?result "dcOpInfo") pv("M4"
"vth" ?result "dcOpInfo") pv("M5" "vth" ?result "dcOpInfo") pv("M6"
"vth" ?result "dcOpInfo") pv("M7" "vth" ?result "dcOpInfo") pv("M8" "vth"
?result "dcOpInfo") ?numberNotation 'scientific)
```

References

[1] "WSTS semiconductor market forecast spring 2017." November 2017. [Online]. Available: www.wsts.org

[2] "Semiconductor shipments forecast to exceed 1 trillion devices in 2018," November 2017. [Online]. Available: www.icinsights.com

[3] G. G. Gielen, "CAD tools for embedded analogue circuits in mixed-signal integrated systems on chip," *IEE Proceedings - Computers and Digital Techniques*, vol. 152, no. 3, pp. 317–332, 2005.

[4] F. M and K. B, "FPAD: a fuzzy nonlinear programming approach to analog circuit design," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 14, no. 7, pp. 785–793, 1995.

[5] A. Jafari, "Artificial neural networks based approach for parametric optimization of cmos operational amplifiers," *International Review on Modelling and Simulations*, vol. 4, pp. 2032–2037, 08 2011.

[6] "Zion Market Research," January 2019. [Online]. Available: www.zionmarketresearch.com

[7] E. S. Ochotta, T. Mukherjee, R. A. Rutenbar, and L. R. Carley, *Practical synthesis of high-performance analog circuits*. Springer Science & Business Media, 2012.

[8] "HSPICE users™ manual release H92," 1992.

[9] "Cadence Design Systems Inc." November 2003. [Online]. Available: www.cadence.com

[10] E. Hennig, "An automated approach for sizing complex analog circuits in a simulation-based flow," *www. neolinear. com*, 2002.

[11] A. H. Shah, S. Dugalleix, and F. Lemery, "Technology migration of a high performance CMOS amplifier using an automated front-to-back analog design flow," in *Design, Automation and Test in Europe Conference and Exhibition, 2002. Proceedings of Designers™ Forum*, 2002, pp. 224–229.

[12] F. V. Fernandez and A. Rodriguez-Vazquez, "Symbolic analysis tools-the state of the art," in *1996 IEEE International Symposium on Circuits and Systems. Circuits and Systems Connecting the World. ISCAS 96*, vol. 4. IEEE, 1996, pp. 798–801.

[13] 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*, vol. 19, no. 6, pp. 703–717, June 2000.

[14] J. Nocedal and S. Wright, "Numerical optimization: Springer science & business media," *New York*, 2006.

[15] B. Liu, G. Gielen, and F. V. Fernández, "Automated design of analog and high-frequency circuits," in *A computational intelligence approach*. Springer, 2014.

[16] G. G. Gielen, H. C. Walscharts, and W. M. Sansen, "Analog circuit design optimization based on symbolic simulation and simulated annealing," *IEEE Journal of Solid-State Circuits*, vol. 25, no. 3, pp. 707–713, 1990.

[17] J. Yuan, N. Farhat, and J. Van der Spiegel, "GBOPCAD: A synthesis tool for high-performance gain-boosted opamp design," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 52, no. 8, pp. 1535–1544, 2005.

[18] L. C. Severo and A. Girardi, "An optimization-based tool for circuit level synthesis analog integrated circuits," in *IX Microelectronics Students Forum, Chip on the Dunes, Natal*, 2009.

[19] J. Bland and G. Dawson, "Tabu search and design optimization," *Computer-aided design*, vol. 23, no. 3, pp. 195–201, 1991.

[20] D. L. W. Kruiskamp, "DARWIN: CMOS opamp synthesis by means of a genetic algorithm," in *32nd Design Automation Conference*. IEEE, 1995, pp. 433–438.

[21] R. Menozzi, A. Piazzi, and F. Contini, "Small-signal modeling for microwave FET linear circuits based on a genetic algorithm," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 43, no. 10, pp. 839–847, 1996.

[22] C. Rajagopal, K. Sridhar, and A. Nunez, "RF CMOS circuit optimizing procedure and synthesis tool," in *Proceedings of the 13th ACM Great Lakes symposium on VLSI*. ACM, 2003, pp. 100–103.

[23] B. Liu, Y. Wang, Z. Yu, L. Liu, M. Li, Z. Wang, J. Lu, and F. V. Fernández, "Analog circuit optimization system based on hybrid evolutionary algorithms," *INTEGRATION, the VLSI journal*, vol. 42, no. 2, pp. 137–148, 2009.

[24] Y. Li, "A simulation-based evolutionary approach to LNA circuit design optimization," *Applied Mathematics and Computation*, vol. 209, no. 1, pp. 57–67, 2009.

[25] M. Barros, J. Guilherme, and N. Horta, "Analog circuits optimization based on evolutionary computation techniques," *Integration*, vol. 43, no. 1, pp. 136–155, 2010.

[26] J. R. Koza, F. H. Bennett, D. Andre, M. A. Keane, and F. Dunlap, "Automated synthesis of analog electrical circuits by means of genetic programming," *IEEE Transactions on evolutionary computation*, vol. 1, no. 2, pp. 109–128, 1997.

[27] J. R. Koza, S. H. Al-Sakran, and L. W. Jones, "Cross-domain features of runs of genetic programming used to evolve designs for analog circuits, optical lens systems, controllers, antennas, mechanical systems, and quantum computing circuits," in *2005 NASA/DoD Conference on Evolvable Hardware (EH'05)*. IEEE, 2005, pp. 205–212.

[28] H. Ryose, K. Jin'no, and H. Hirose, "Automated synthesis of simple nonlinear analog circuits by means of genetic algorithm," *Journal of Signal Processing*, vol. 8, no. 6, pp. 529–535, 2004.

[29] F. Wang, Y. Li, L. Li, and K. Li, "Automated analog circuit design using two-layer genetic programming," *Applied mathematics and computation*, vol. 185, no. 2, pp. 1087–1097, 2007.

[30] K. Zielinski and R. Laur, "Constrained single-objective optimization using differential evolution," in *2006 IEEE International Conference on Evolutionary Computation*. IEEE, 2006, pp. 223–230.

[31] M. KUBAŘ and J. JAKOVENKO, "A powerful optimization tool for analog integrated circuits design." *Radioengineering*, vol. 22, no. 3, 2013.

[32] A. Kalinli, "Optimal circuit design using immune algorithm," in *International Conference on Artificial Immune Systems*. Springer, 2004, pp. 42–52.

[33] H. Xu and Y. Ding, "Optimizing method for analog circuit design using adaptive immune genetic algorithm," in *2009 Fourth International Conference on Frontier of Computer Science and Technology*. IEEE, 2009, pp. 359–363.

[34] M. R. Tamplin and A. Hamilton, "Ant circuit world: An ant algorithm MATLAB toolbox for the design, visualisation and analysis of analogue circuits," in *International Conference on Evolvable Systems*. Springer, 2001, pp. 151–158.

[35] B. Benhala, A. Ali, A. Mechaqrane, B. Benlahbib, F. Abdi, E. Hossain Abarkan, and M. Fakhfakh, "Sizing of current conveyors by means of an ant colony optimization technique," 05 2011, pp. 1 – 6.

[36] B. Benhala, A. Ahaitouf, A. Mechaqrane, and B. Benlahbib, "Multi-objective optimization of second generation current conveyors by the ACO technique," *2012 International Conference on Multimedia Computing and Systems*, pp. 1147–1151, 2012.

[37] J. Park, K. Choi, and D. J. Allstot, "Parasitic-aware RF circuit design and optimization," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 51, no. 10, pp. 1953–1966, 2004.

[38] S. K. Mandal, S. Sural, and A. Patra, "ANN-and PSO-based synthesis of on-chip spiral inductors for RF ICs," *IEEE transactions on computer-aided design of integrated circuits and systems*, vol. 27, no. 1, pp. 188–192, 2007.

[39] M. Fakhfakh, S. Masmoudi, Y. Cooren, M. Loulou, and P. Siarry, "Improving switched current sigma delta modulators performances via the particle swarm optimization technique," in *Modeling, Analysis, and Applications in Metaheuristic Computing: Advancements and Trends*. IGI Global, 2012, pp. 154–170.

[40] A. Sallem, B. Benhala, M. Kotti, M. Fakhfakh, A. Ahaitouf, and M. Loulou, "Application of swarm intelligence techniques to the design of analog circuits: evaluation and comparison," *Analog Integrated Circuits and Signal Processing*, vol. 75, no. 3, pp. 499–516, 2013.

[41] A. El Dor, M. Fakhfakh, and P. Siarry, "Performance optimization of CMOS second generation current conveyors using a multi-swarm algorithm," *AEU-International Journal of Electronics and Communications*, vol. 68, no. 6, pp. 496–503, 2014.

[42] M. Fakhfakh, Y. Cooren, A. Sallem, M. Loulou, and P. Siarry, "Analog circuit design optimization through the particle swarm optimization technique," *Analog integrated circuits and signal processing*, vol. 63, no. 1, pp. 71–82, 2010.

[43] P. Kumar and K. Duraiswamy, "An optimized device sizing of analog circuits using particle swarm optimization," in *Proceedings of IEEE International Conference on Neural Networks, Nov. 27-Dec. 1, IEEE Xplore*. Citeseer, 2012.

[44] S. Mallick, R. Kar, D. Mandal, and S. Ghoshal, "Optimal sizing of CMOS analog circuits using gravitational search algorithm with particle swarm optimization," *International Journal of Machine Learning and Cybernetics*, vol. 8, no. 1, pp. 309–331, 2017.

[45] M. G. Degrauwe, O. Nys, E. Dijkstra, J. Rijmenants, S. Bitz, B. L. Goffart, E. A. Vittoz, S. Cserveny, C. Meixenberger, G. Van Der Stappen *et al.*, "IDAC: An interactive design tool for analog CMOS circuits," *IEEE Journal of solid-state circuits*, vol. 22, no. 6, pp. 1106–1116, 1987.

[46] R. Harjani, R. A. Rutenbar, and L. R. Carley, "Analog circuit synthesis and exploration in OASYS," in *Proceedings 1988 IEEE International Conference on Computer Design: VLSI*. IEEE, 1988, pp. 44–47.

[47] R. Harjani, R. A. Rutenbar, and L. R. Carley, "OASYS: a framework for analog circuit synthesis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 8, no. 12, pp. 1247–1266, Dec 1989.

[48] F. El-Turky and E. E. Perry, "BLADES: an artificial intelligence approach to analog circuit design," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 8, no. 6, pp. 680–692, June 1989.

[49] C. A. Makris and C. Toumazou, "Analog IC design automation. II. automated circuit correction by qualitative reasoning," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 14, no. 2, pp. 239–254, Feb 1995.

[50] G. Alpaydin, S. Balkir, and G. Dundar, "An evolutionary approach to automatic synthesis of high-performance analog integrated circuits," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 3, pp. 240–252, June 2003.

[51] S. Balkar, G. Dundar, and A. S. Ogrenci, "Analog VLSI design automation," p. 218 pages :. [Online]. Available: http://olin.tind.io/record/131716

[52] K. Mendhurwar, H. Sundani, P. Aggarwal, R. Raut, and V. Devabhaktuni, "A new approach to sizing analog CMOS building blocks using pre-compiled neural network models," *Analog Integrated Circuits and Signal Processing*, vol. 70, no. 3, pp. 265–281, 2012.

[53] N. Kahraman and T. Yildirim, "Technology independent circuit sizing for fundamental analog circuits using artificial neural networks," in *2008 Ph.D. Research in Microelectronics and Electronics*, June 2008, pp. 1–4.

[54] N. Kahraman and T. Yildirim, "Technology independent circuit sizing for standard cell based design using neural networks," *Digital Signal Processing*, vol. 19, no. 4, pp. 708–714, 2009.

[55] A. Andreev, V. Kazakov, and A. Svetlov, "A macromodel of an operational amplifier for modelling measurement circuits with pulsed signals," *Measurement Techniques*, vol. 42, no. 9, pp. 858–863, 1999.

[56] S. Balkir, G. Dundar, and G. Alpaydin, "Evolution based synthesis of analog integrated circuits and systems," in *Proceedings. 2004 NASA/DoD Conference on Evolvable Hardware, 2004.* IEEE, 2004, pp. 26–29.

[57] H. Y. Koh, C. H. Sequin, and P. R. Gray, "OPASYN: A compiler for cmos operational amplifiers," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 9, no. 2, pp. 113–125, 1990.

[58] J. P. Harvey, M. I. Elmasry, and B. Leung, "STAIC: An interactive framework for synthesizing cmos and bicmos analog circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 11, no. 11, pp. 1402–1417, 1992.

[59] P. C. Maulik and L. R. Carley, "Automating analog circuit design using constrained optimization techniques," in *1991 IEEE International Conference on Computer-Aided Design Digest of Technical Papers*. IEEE, 1991, pp. 390–393.

[60] P. C. Maulik, L. R. Carley, and R. A. Rutenbar, "A mixed-integer nonlinear programming approach to analog circuit synthesis," in *1992 Proceedings 29th ACM/IEEE Design Automation Conference*. IEEE, 1992, pp. 698–703.

[61] Y. Cheng and C. Hu, *MOSFET modeling & BSIM3 user™ guide*. Springer Science & Business Media, 1999.

[62] W. LIU and C. HU, "Bsim3v3 mosfet model," *International Journal of High Speed Electronics and Systems*, vol. 09, no. 03, pp. 671–701, 1998.

[63] E. S. Ochotta, R. A. Rutenbar, and L. R. Carley, "Synthesis of high-performance analog circuits in ASTRX/OBLX," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 15, no. 3, pp. 273–294, 1996.

[64] G. Van der Plas, G. Debyser, F. Leyn, K. Lampaert, J. Vandenbussche, G. G. Gielen, W. Sansen, P. Veselinovic, and D. Leenarts, "AMGIE-A synthesis environment for CMOS analog integrated circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 20, no. 9, pp. 1037–1058, 2001.

[65] G. Van der Plas, G. Gielen, and W. M. Sansen, *A computer-aided design and synthesis environment for analog integrated circuits*. Springer Science & Business Media, 2006, vol. 672.

[66] M. del Mar Hershenson, S. P. Boyd, and T. H. Lee, "GPCAD: a tool for cmos op-amp synthesis," in *1998 IEEE/ACM International Conference on Computer-Aided Design. Digest of Technical Papers (IEEE Cat. No.98CB36287)*, Nov 1998, pp. 296–303.

[67] M. d. Hershenson, S. P. Boyd, and T. H. Lee, "Optimal design of a CMOS op-amp via geometric programming," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 20, no. 1, pp. 1–21, Jan 2001.

[68] F. Medeiro, B. Perez-Verdu, A. Rodriguez-Vazquez, and J. L. Huertas, "A vertically integrated tool for automated design of $\sigma\delta$ modulators," *IEEE Journal of Solid-State Circuits*, vol. 30, no. 7, pp. 762–772, 1995.

[69] A. Doboli, N. Dhanwada, A. Nunez-Aldana, and R. Vemuri, "A two-layer library-based approach to synthesis of analog systems from VHDL-AMS specifications," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 9, no. 2, pp. 238–271, Apr. 2004. [Online]. Available: http://doi.acm.org/10.1145/989995.990000

[70] K. Matsukawa, T. Morie, Y. Tokunaga, S. Sakiyama, Y. Mitani, M. Takayama, T. Miki, A. Matsumoto, K. Obata, and S. Dosho, "Design methods for pipeline & delta-sigma A-to-D converters with convex optimization," in *Proceedings of the 2009 Asia and South Pacific Design Automation Conference*, ser. ASP-DAC '09. Piscataway, NJ, USA: IEEE Press, 2009, pp. 690–695.

[71] S.-H. Lui, H.-K. Kwan, and N. Wong, "Analog circuit design by nonconvex polynomial optimization: Two design examples," *International Journal of Circuit Theory and Applications*, vol. 38, no. 1, pp. 25–43, 2010.

[72] M. Fakhfakh, Y. Cooren, A. Sallem, M. Loulou, and P. Siarry, "Analog circuit design optimization through the particle swarm optimization technique," *Analog Integrated Circuits and Signal Processing*, vol. 63, no. 1, pp. 71–82, Apr 2010. [Online]. Available: https://doi.org/10.1007/s10470-009-9361-3

[73] M. Kotti, B. Benhala, M. Fakhfakh, A. Ahaitouf, B. Benlahbib, M. Loulou, and A. Mecheqrane, "Comparison between PSO and ACO techniques for analog circuit performance optimization," in *ICM 2011 Proceeding*. IEEE, 2011, pp. 1–6.

[74] R. A. Vural, T. Yildirim, T. Kadioglu, and A. Basargan, "Performance evaluation of evolutionary algorithms for optimal filter design," *IEEE transactions on evolutionary computation*, vol. 16, no. 1, pp. 135–147, 2011.

[75] R. A. Vural, U. Bozkurt, and T. Yildirim, "Analog active filter component selection with nature inspired metaheuristics," *AEU-International Journal of Electronics and Communications*, vol. 67, no. 3, pp. 197–205, 2013.

[76] R. Vural and T. Yildirim, "Analog circuit sizing via swarm intelligence," *AEU-International journal of electronics and communications*, vol. 66, no. 9, pp. 732–740, 2012.

[77] R. A. Vural, O. Der, and T. Yildirim, "Investigation of particle swarm optimization for switching characterization of inverter design," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5696–5703, 2011.

[78] S. Dam and P. Mandal, "Modeling and design of CMOS analog circuits through hierarchical abstraction," *Integration, the VLSI Journal*, vol. 46, no. 4, pp. 449–462, 2013.

[79] S. Mallick, R. Kar, D. Mandal, and S. Ghoshal, "Optimal sizing of CMOS analog circuits using gravitational search algorithm with particle swarm optimization," *International Journal of Machine Learning and Cybernetics*, vol. 8, no. 1, pp. 309–331, 2017.

[80] S. Mallick, R. Kar, D. Mandal, and S. P. Ghoshal, "CMOS analogue amplifier circuits optimisation using hybrid backtracking search algorithm with differential evolution," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 28, no. 4, pp. 719–749, 2016.

[81] B. P. De, R. Kar, D. Mandal, and S. P. Ghoshal, "An efficient design of CMOS comparator and folded cascode op-amp circuits using particle swarm optimization with an aging leader and challengers algorithm," *International Journal of Machine Learning and Cybernetics*, vol. 7, no. 2, pp. 325–344, 2016.

[82] B. P. De, R. Kar, D. Mandal, and S. Ghoshal, "Soft computing-based approach for optimal design of on-chip comparator and folded-cascode op-amp using colliding bodies optimization," *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, vol. 29, no. 5, pp. 873–896, 2016.

[83] S. Mallick, R. Kar, S. Ghoshal, and D. Mandal, "Optimal sizing and design of CMOS analogue amplifier circuits using craziness-based particle swarm optimization," *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, vol. 29, no. 5, pp. 943–966, 2016.

[84] K. B. Maji, R. Kar, D. Mandal, and S. Ghoshal, "Optimal design of low power high gain and high speed CMOS circuits using fish swarm optimization algorithm," *International Journal of Machine Learning and Cybernetics*, vol. 9, no. 5, pp. 771–786, 2018.

[85] K. Maji, R. Kar, D. Mandal, and S. Ghoshal, "An evolutionary approach based design automation of low power CMOS two-stage comparator and folded cascode OTA," *AEU-International Journal of Electronics and Communications*, vol. 70, no. 4, pp. 398–408, 2016.

[86] A. B. de Andrade, A. Petraglia, and C. F. Soares, "A constrained optimization approach for accurate and area efficient bandgap reference design," *Microelectronics journal*, vol. 65, pp. 72–77, 2017.

[87] C. L. Singh, C. Anandini, A. J. Gogoi, and K. Baishnab, "Automated sizing of low-noise CMOS analog amplifier using ALCPSO optimization algorithm," *Journal of Information and Optimization Sciences*, vol. 39, no. 1, pp. 99–111, 2018.

[88] B. P. De, K. B. Maji, R. Kar, D. Mandal, and S. P. Ghoshal, "Evolutionary computation based sizing technique of nulling resistor compensation based cmos two-stage op-amp circuit," *International Journal of High Speed Electronics and Systems*, vol. 26, no. 04, p. 1740021, 2017.

[89] B. De, K. Maji, R. Kar, D. Mandal, and S. Ghoshal, "Design of optimal CMOS analog amplifier circuits using a hybrid evolutionary optimization technique," *Journal of Circuits, Systems and Computers*, vol. 27, no. 02, p. 1850029, 2018.

[90] P. Sarkar, N. M. Laskar, S. Nath, K. Baishnab, and S. Chanda, "Offset voltage minimization based circuit sizing of CMOS operational amplifier using whale optimization algorithm," *Journal of Information and Optimization Sciences*, vol. 39, no. 1, pp. 83–98, 2018.

[91] N. M. Laskar, K. Guha, I. Chatterjee, S. Chanda, K. L. Baishnab, and P. K. Paul, "HW-PSO: A new hybrid whale-particle swarm optimization algorithm and its application in electronic design optimization problems," *Applied Intelligence*, vol. 49, no. 1, pp. 265–291, 2019.

[92] G. Neves, M. Barros, and N. Horta, "AIDA: Analog IC design automation based on a fully configurable design hierarchy and flow," in *2006 13th IEEE International Conference on Electronics, Circuits and Systems*. IEEE, 2006, pp. 490–493.

[93] W. Nye, D. C. Riley, A. Sangiovanni-Vincentelli, and A. L. Tits, "DELIGHT. SPICE: An optimization-based system for the design of integrated circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 7, no. 4, pp. 501–519, 1988.

[94] K.-H. Meng, P.-C. Pan, and H.-M. Chen, "Integrated hierarchical synthesis of analog/RF circuits with accurate performance mapping," in *2011 12th International Symposium on Quality Electronic Design*. IEEE, 2011, pp. 1–8.

[95] A. J. Torralba, J. Chávez, and L. G. Franquelo, "Fuzzy-logic-based analog design tools," *IEEE Micro*, vol. 16, no. 4, pp. 60–68, 1996.

[96] A. Torralba, J. Chavez, and L. G. Franquelo, "FASY: a fuzzy-logic based tool for analog synthesis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 15, no. 7, pp. 705–715, 1996.

[97] C.-W. Lin, P.-D. Sue, Y.-T. Shyu, and S.-J. Chang, "A bias-driven approach for automated design of operational amplifiers," in *2009 International Symposium on VLSI Design, Automation and Test*. IEEE, 2009, pp. 118–121.

[98] F. Medeiro, F. V. Fernández, R. Domínguez-Castro, and A. Rodriguez-Vazquez, "A statistical optimization-based approach for automated sizing of analog cells," in *Proceedings of the 1994 IEEE/ACM international conference on Computer-aided design*. IEEE Computer Society Press, 1994, pp. 594–597.

[99] R. Castro-Lopez, O. Guerra, E. Roca, and F. V. Fernández, "An integrated layout-synthesis approach for analog ICs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 7, pp. 1179–1189, 2008.

[100] M. F. Barros, J. M. Guilherme, and N. C. Horta, *Analog circuits and systems optimization based on evolutionary computation techniques*. Springer, 2010, vol. 9.

[101] M. Barros, J. Guilherme, and N. Horta, "Analog circuits optimization based on evolutionary computation techniques," *Integration*, vol. 43, no. 1, pp. 136–155, 2010.

[102] M. Barros, J. Guilherme, and N. 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*. ACM, 2007, pp. 469–472.

[103] R. Santos-Tavares, N. Paulino, J. Higino, J. Goes, and J. P. Oliveira, "Optimization of multi-stage amplifiers in deep-submicron CMOS using a distributed/parallel genetic algorithm," in *2008 IEEE International Symposium on Circuits and Systems*. IEEE, 2008, pp. 724–727.

[104] M. Krasnicki, R. Phelps, R. A. Rutenbar, and L. R. Carley, "MAELSTROM: efficient simulation-based synthesis for custom analog cells," in *Proceedings 1999 Design Automation Conference (Cat. No. 99CH36361)*. IEEE, 1999, pp. 945–950.

[105] A. Jafari, E. Bijami, H. R. Bana, and S. Sadri, "A design automation system for CMOS analog integrated circuits using new hybrid shuffled frog leaping algorithm," *Microelectronics Journal*, vol. 43, no. 11, pp. 908–915, 2012.

[106] M. KUBAR and J. JAKOVENKO, "A powerful optimization tool for analog integrated circuits design." *Radioengineering*, vol. 22, no. 3, 2013.

[107] M. Barari, H. R. Karimi, and F. Razaghian, "Analog circuit design optimization based on evolutionary algorithms," *Mathematical Problems in Engineering*, vol. 2014, 2014.

[108] M. Dehbashian and M. Maymandi-Nejad, "A new hybrid algorithm for analog ICs optimization based on the shrinking circles technique," *Integration*, vol. 56, pp. 148–166, 2017.

[109] Maryam Dehbashian and Mohammad Maymandi Nejad, "An enhanced optimization kernel for analog IC design automation using the shrinking circles technique," *Engineering Applications of Artificial Intelligence*, vol. 58, pp. 62 – 78, 2017.

[110] (2003) Neolinear inc. [Online]. Available: http://www.neolinear.com

[111] "Analog Design Automation Inc." June 2003. [Online]. Available: homepage: www.analogsynthesis.com

[112] (2003) Barcelona design. [Online]. Available: http://www.barcelonadesign.com

[113] (2003) Anasift. [Online]. Available: http://www.anasift.com

[114] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46 – 61, 2014.

[115] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51 – 67, 2016.

[116] J. A. Goldbogen, A. S. Friedlaender, J. Calambokidis, M. F. McKenna, M. Simon, and D. P. Nowacek, "Integrative Approaches to the Study of Baleen Whale Diving Behavior, Feeding Performance, and Foraging Ecology," *BioScience*, vol. 63, no. 2, pp. 90–100, 02 2013.

[117] P. E. Allen and D. R. Holberg, *CMOS Analog Circuit Design*. Oxford University Press, 2002.

[118] M. Dehbashian and M. Maymandi-Nejad, "An enhanced optimization kernel for analog IC design automation using the shrinking circles technique," *Engineering Applications of Artificial Intelligence*, vol. 58, pp. 62–78, Feb. 2017.

[119] P. P. Prajapati and M. V. Shah, "Two stage CMOS operational amplifier design using particle swarm optimization algorithm," in *2015 IEEE UP Section Conference on Electrical Computer and Electronics (UPCON)*, Dec 2015, pp. 1–5.

[120] P. Sarkar, N. M. Laskar, S. Nath, K. L. Baishnab, and S. Chanda, "Offset voltage minimization based circuit sizing of CMOS operational amplifier using whale optimization algorithm," *Journal of Information and Optimization Sciences*, vol. 39, no. 1, pp. 83–98, 2018.

[121] B. P. De, K. B. Maji, R. kar, D. Mandal, and S. P. Ghosal, "Evolutionary computation based sizing technique of nulling resistor compensation based CMOS two-stage Op-Amp circuit." *International Journal of high Speed Electronics and Systems*, vol. 26, no. 04, p. 1740021, 2017.

[122] W. M. C. Sansen, *Analog Design Essentials*. Springer, 2006.

[123] Z. Yu, X. Yang, and S. Chung, "A compact operational amplifier with load-insensitive stability compensation for high-precision transducer interface," *Sensors*, vol. 18, no. 2, January 2018.

[124] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, April 1997.

[125] H. Nenavath and R. K. Jatoth, "Hybridizing sine cosine algorithm with differential evolution for global optimization and object tracking," *Applied Soft Computing*, vol. 62, pp. 1019 – 1043, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1568494617305781

[126] S. Mirjalili, "SCA: A sine cosine algorithm for solving optimization problems," *Knowledge-Based Systems*, vol. 96, pp. 120 – 133, 2016.

[127] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, Nov 1995, pp. 1942–1948 vol.4.

[128] N. Mittal, U. Singh, and B. S. Sohi, "Modified grey wolf optimizer for global engineering optimization," *Applied Computational Intelligence and Soft Computing*, vol. 2016, p. 16, 2016.

[129] E.-G. Talbi, "A taxonomy of hybrid metaheuristics," *Journal of Heuristics*, vol. 8, no. 5, p. 541–564, 2002.

[130] S. Mirjalili and S. Z. M. Hashim, "A new hybrid PSOGSA algorithm for function optimization," in *2010 International Conference on Computer and Information Application*, Dec 2010, pp. 374–377.

[131] B. Liu, Y. Wang, Z. Yu, L. Liu, M. Li, Z. Wang, J. Lu, and F. V. Fernandez, "Analog circuit optimization system based on hybrid evolutionary algorithms," *Integration*, vol. 42, no. 2, pp. 137 – 148, 2009.

[132] G. Singh, "Design and analysis of CMOS folded cascode OTA using gm/ID technique," *International Journal of Electronics and Computer Science Engineering (IJECSE)*, vol. 1, pp. 727–733, 01 2012.

[133] H. Daoud Dammak, S. Bensalem, S. Zouari, and M. Loulou, "Design of folded cascode OTA in different regions of operation through gm/ID methodology," *International Journal of Electrical and Electronics Engineering*, vol. 1, 01 2008.

[134] B. Samir, B. Lakehal, A. Dendouga, S. Kouda, and N.-E. Bouguechal, "Optimization of fully-differential folded cascode amplifier via multi-objective genetic algorithms," in *The International Conference on Electronics & Oil: From Theory to Applications*, 03 2013.

[135] S. Alapati and S. Patri, "Improved transient response capacitor less low dropout regulator employing adaptive bias and bulk modulation," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 26, pp. 2385–2394, 09 2018.

[136] A. Maity and A. Patra, "A single-stage low-dropout regulator with a wide dynamic range for generic applications," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 6, pp. 2117–2127, June 2016.

List of Publications

International Journals

1. M A Mushahhid Majeed, P Sreehari Rao, “An enhanced grey wolf optimization algorithm with improved exploration ability for analog circuit design automation", Turkish Journal of Electrical Engineering & Computer Sciences 2018, 26: 2605 - 2617.

2. M.A. Mushahhid Majeed, Sreehari Rao Patri, (2018) "A hybrid of WOA and mGWO algorithms for global optimization and analog circuit design automation", COMPEL - The international journal for computation and mathematics in electrical and electronic engineering.

3. M A Mushahhid Majeed, Sreehari P, “Optimal Design of CMOS Analog Circuit Using Enhanced Grey Wolf Optimization Algorithm”, Journal of Advanced Research in Dynamical and Control Systems, 2018.

4. M A Mushahhid Majeed, Sreehari Rao P, “Optimal Design of CMOS Amplifier Circuits Using Whale Optimization Algorithm” International Conference on Communication, Networks and Computing, Communications in Computer and Information Science - Springer, 2018.

5. M A Mushahhid Majeed, Sreehari Rao P, “Process-independant Simulation-based Design Automation of CMOS Analog Circuits Using Metaheuristic Algorithms” Applied Soft Computing, Elsevier. (Under Review)

International conferences

1. M. A. Mushahhid Majeed and P. S. Rao, "Optimization of CMOS analog circuits using sine cosine algorithm," IEEE 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Delhi, 2017, pp. 1-6.

2. M. A. Mushahhid Majeed and P. S. Rao, "Optimization of CMOS Analog Circuits Using Grey Wolf Optimization Algorithm," 2017 14th IEEE India Council International Conference (INDICON), Roorkee, 2017, pp. 1-6.

3. M A Mushahhid Majeed, Sreehari Rao P, “Optimal Circuit Sizing of CMOS Operational Amplifier Using Modified Grey Wolf Optimization Algorithm”, IEEE-International Conference on Innovative Technologies in Engineering 2018, Hyderabad. (Accepted)