Novel Metaheuristics for the Performance Analysis and the Design Optimization of VLSI Circuits

A

Thesis submitted

for the award of the degree of

DOCTOR OF PHILOSOPHY

By

SATYABRATA DASH

DEPARTMENT OF ELECTRONICS AND ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

GUWAHATI - 781 039, ASSAM, INDIA

OCT 2018

Certificate

This is to certify that the thesis entitled "**Novel Metaheuristics for the Performance Analysis and the Design Optimization of VLSI Circuits**", submitted by **SATYABRATA DASH** (11610242), a research scholar in the *Department of Electronics and Electrical Engineering, Indian Institute of Technology Guwahati*, for the award of the degree of **Doctor of Philosophy**, is a record of an original research work carried out by him under my supervision and guidance. The thesis has fulfilled all requirements as per the regulations of the institute and in my opinion has reached the standard needed for submission. The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Date:

Place : Guwahati

Dr. Gaurav Trivedi
Associate Professor
Dept. of Electronics and Electrical Engg.
Indian Institute of Technology Guwahati
Guwahati - 781 039, Assam, India.

“Om Namo Bhagavate Sri Aurobindaya ”

Dedicated to my **parents** and **brother**

for their blessings, love and support.

Acknowledgement

Meagre expression conceals major part of my cardiac gratitude and indebtedness for my research supervisor, Dr. Gaurav Trivedi, a distinguished conceptualist, prolific trailblazer, not out of sheer imposition for mere duty's sake; but for his heraldic highlight, extolling suggestions, constant guidance, timely advice, scrupulous scrutiny and explicit encouragement.

I would like to extend my profound sense of gratitude to Prof. P. K. Bora, Dr. S. R. Ahamed and Dr. A. Sahu for thorough inspiration and valuable suggestions. I am also grateful to faculty members and the office staffs of the Department of Electronics and Electrical Engineering, IIT Guwahati, for their help in carrying out this research work.

I also express my sincere and deep sense of gratitude to Prof. S. Patkar, IIT Bombay, for his scholastic advices, benevolent and many fold helps in connection with this research project. I am thankful to Mr. Surinder Singh, Mr. H. S. Jatana, Mr. Deep Sehgal and Mr. Ashutosh, Semiconductor Lab, Chandigarh for their kind support in providing industry benchmarks to conduct the research work.

My sincere thanks go to Mr. Deepak Joshi and Mr. Sukanta Dey for their assistance in writing of the thesis. I am thankful to Dr. Pradeep Kumar Biswal, Dr. Dheeraj Sinha, Dr. Nagaraj Adiga, Mr. Gaurav Kumar, Mr. Sunil Dutt, Mr. Ashok Kumar Ray, Mr. Kashyap Kumar Prabhakar, Mr. Umesh Choudhary and Mr. Anish Augustine for their ungrudging help and sympathetic encouragement.

Last but most important are my parents and other family members whose blessings and love made my path of success. I attribute this achievement to my father, mother, uncle, aunt, brothers and sisters for their constant support, silent prayers for my success and moreover, making me stand in this position.

SATYABRATA DASH

Abstract

Power has become imperative to the design closure in today's ultra low submicron design. The ongoing trends in technology scaling imply to design fast and power efficient circuits. Further, the performance and reliability of integrated circuits have become increasingly susceptible to the aftermath of low supply voltage levels with smaller feature sizes. The impact of reduction in supply voltage levels is multi-discipline to researchers ranging from power supply design, power converters or voltage regulators design, thermal analysis of systems, power distribution network design, signal integrity analysis, digital and analog circuit analysis and sizing, realization of neuromorphic systems, etc. to minimizing power itself. It has become a challenging task to perform various analysis and design tasks using efficient computing environment to analyze the tradeoff between increased on-chip power consumption and low power design. Although various EDA tools are available commercially, there is a growing demand to incorporate new efficient techniques to bridge the gap between increased technology scaling and low power design. Metaheuristics for circuit design can be treated as one of the viable options of the suitable techniques to realize various VLSI circuits while being used in conjunction with current EDA tools. In view of this, three different VLSI circuits, such as power distribution network, analog/RF circuit and memristor crossbar array, are considered in this thesis to be explored for analysis and optimization by several novel metaheuristics.

Due to lower supply voltage levels, the integrated circuits have become more sensitive to power supply noises, such as IR drops, $L(di/dt)$ noises, etc. in on-chip power distribution networks. Further, with extremely large size of power distribution networks, the realistic simulation of such large networks is computationally intensive both in terms of runtime and

memory. In order to analyze large power distribution networks accurately and efficiently, two metaheuristics based on random walk and river formation dynamics are presented in the thesis. Experimental results show that the proposed metaheuristics accelerate the analysis on GPU to achieve remarkable speedups with acceptable accuracy loss. Further, designing a power distribution network is imperative to the VLSI circuit design flow to avoid the occurrence of any major power distribution network problems, such as IR drop and to optimize area (reduced metal area and reduced routable space) for incorporation of other system components. In order to address the above mentioned design problems, we demonstrate the application of river formation dynamics based metaheuristics (RFD and MRFD[1]) to optimize the power distribution network resulting from very large-scale designs. A number of experiments are performed on various power distribution benchmarks to demonstrate the effectiveness of RFD and MRFD in minimizing IR drop and wire area. It is observed that both the metaheuristics show competitive performance with respect to other standard optimization techniques.

With scaling of the technology, the electronic circuit designs have become more complex. The cost of designing such complex systems drives towards the path of employing automation in the field of analog circuit design. Therefore, in this thesis, the applications of both RFD and MRFD metaheuristics are extended to design analog/RF circuits (a two-stage CMOS operational amplifier circuit and a low noise amplifier circuit) with user-defined specifications. Various experiments are performed to evaluate these analog/RF circuits and it is observed that both the metaheuristics show competitive performance in terms of quality of solutions when compared with standard optimization techniques. Further, as circuit design process requires the need to analyze tradeoffs among various design specifications, competing objective functions (two or more contradictory design specifications) can be analyzed simultaneously to capture the performance of a circuit. In this regard, we propose three different multiobjective optimization algorithms[2], i.e., hNSGA-II, MOMRFD and IMBSO to find a set of nondominated solutions (Pareto optimal solutions) among different performance specifi-

[1]RFD = River Formation Dynamics, MRFD = Modified River Formation Dynamics

[2]hNSGA-II = Hierarchical Nondominated Sorting Genetic Algorithm-II, MOMRFD = Multiobjective Modified River Formation Dynamics, IMBSO = Improved Brain Storm Optimization

cations of a two-stage operational amplifier circuit, a folded cascode amplifier circuit and a low noise amplifier circuit. It is observed that IMBSO algorithm outperforms the proposed techniques and other peer algorithms in terms of convergence and diversity of the solutions.

Finally, the application of both RFD and MRFD metaheuristics is extended to minimize IR drop in memristor crossbar arrays. IR drop problem in memristor crossbar array is transformed into a single objective constrained optimization problem subjected to a set of design constraints, i.e., crossbar size, IR drop threshold and wire width. To demonstrate the improvement, various experiments have been performed on various memristor crossbar array benchmarks for a number of function evaluations and the results are compared with other peer algorithms.

Contents

List of Figures

List of Tables

List of Acronyms

ALC-PSO	Particle Swarm Optimization with Aging Leader and Challengers
BSO	Brain Storm Optimization
CAM	Computer Aided Manufacturing
CMA-ES	Covariance Matrix Adaptation Evolution Strategy
CMOS	Complementary Metal Oxide Semiconductor
CRPSO	Craziness based Particle Swarm Optimization
CUDA	Compute Unified Device Architecture
DE	Differential Evolution
dMOPSO	Decomposition based Multiobjective Particle Swarm Optimization
EDA	Electronics Design Automation
FE	Function Evaluation
GA	Genetic Algorithm
GBW	Gain Bandwidth Product
GD	Generational Distance
GMRES	Generalized Minimal Residual Method
GND	Ground
GPU	Graphics Processing Unit
hNSGA-II	Hierarchical Nondominated Sorting Genetic Algorithm II
HV	Hypervolume
IC	Integrated Circuit
ICMR	Input Common Mode Ratio

IGD	Inverse Generational Distance
IMBSO	Improved Multiobjective Brain Storm Optimization
IR	Current(I) $\times$ resistance(R)
ITRS	International Technology Roadmap for Semiconductors
JADE	Adaptive Differential Evolution With Optional External Archive
Levy-PSO	Levy flight based Particle Swarm Optimization
LNA	Low Noise Amplifier
LRT	Likelihood Ratio Test
LSHADE	Success History based Adaptive Differential Evolution with Linear population size reduction
LU	Lower Upper
MCA	Memristor Crossbar Array
MCD	Modified Crowding Distance
MOPSO	Multiobjective Particle Swarm Optimization
MOR	Model Order Reduction
MOS	Metal Oxide Semiconductor
MOMRFD	Multiobjective Modified River Formation Dynamics
MRFD	Modified River Formation Dynamics
NCS	Neuromorphic Compting System
NF	Noise Figure
NP	Nondeterministic Polynomial time
NSGA-II	Nondominated Sorting Genetic Algorithm II
NVCC	Nvidia CUDA Compiler
PCG	Preconditioned Conjugate Gradient
PDN	Power Distribution Network
PF	Pareto Front
PM	Phase Margin
PSO	Particle Swarm Optimization

RFD	River Formation Dynamics
SADE	Self-Adaptive Differential Evolution
SMPSO	Speed-constrained Multiobjective Particle Swarm Optimization
SPICE	Simulation Program with Integrated Circuit Emphasis
SP	Success Performance
SPSO	Standard Particle Swarm Optimization
TSRW	Two-Step Random Walk
UMOEA-II	United Multi-operator Evolutionary Algorithm II
UGW	Unity Gain Bandwidth
VLSI	Very Large Scale Integration

1

Introduction

Contents

1.1 Evolution in integrated circuit technology and design trends

From the invention of the first point-contact transistor at Bell Labs to the availability of first commercial integrated circuit (IC), people could not anticipate the broad social impact that this invention would produce. The prediction that the integration density, i.e., the number of transistors on an integrated circuit leading to the minimum cost per integrated component, will continue to double every year is proved to be correct not only until 1975 but till today.

> *The complexity for minimum component costs has increased at a rate of roughly a factor of two per year... Over the longer term, the rate of increase is a bit more uncertain, although there is no reason to believe it will not remain nearly constant for at least 10 years. That means by 1975, the number of components per integrated circuit for minimum cost will be 65,000. I believe that such a large circuit can be built on a single wafer.*
>
> – G. Moore, 1965 *[1]*

The spectacular improvement in circuit complexity and performance is due to the steady decrease in feature size of semiconductor devices. With the advances in lithographic technology, it is possible to manufacture on-chip structures with higher resolution. As a result, the area, speed and power characteristics of transistors improve with scaling in lateral dimensions of devices. However, with the widespread use of bipolar devices in the development of ICs, many obstacles preclude the scaling of ICs, e.g., (i) on-chip resistors and diodes make inefficient use of die area, (ii) the speed of bipolar transistors does not scale with the lateral dimensions, (iii) with difficulty in device isolation and inefficient design, power consumption increases, etc. Further, with the maturity in metal oxide semiconductor (MOS) technology, the scaling potential of MOS devices is realized. The compactness of MOS circuits and higher yield capability resulted in dramatic increase in density of devices per IC as compared to bipolar ICs. Alternatively, it can be presumed in an ironic way that the refinement of bipolar technology has paved the path to the efficient exploitation of MOS technology and rapid development of high density ICs.

With seminal advances in fabrication technology, there has been an explosive growth of microelectronics industry. The emergence of new applications has imposed various challenges on principal objectives of designing ICs over the past three decades. Further, these challenges have induced several shifts to the IC design paradigm. The evolution of IC design paradigm is shown in Figure 1.1. In the 1960's and 1970's, the primary concern was to maximize yield of such high density ICs by adding the challenges to design more compact circuits and die area. As the electronic systems were made by incorporating more number of ICs, the interconnects were supported by both intra-chip and the board level interconnects. However, as board level interconnects have high latency and dissipate large amounts of power, it limits the speed and power performances of the system. This favors the development of higher functionality per silicon area ICs with ensuing reduction in the number of ICs comprising a system. With lower number of ICs, the speed performance of a system became increasingly dependent on the speed of IC. Therefore, the development of high speed ICs became a primary design objective by the 1980's.

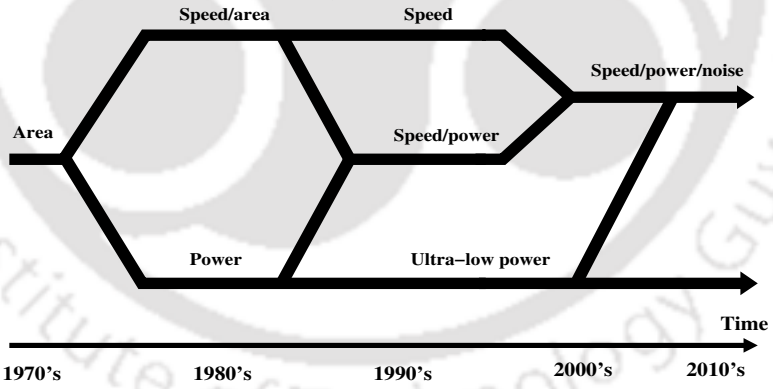


Figure 1.1: Evolution of integrated circuit design trend [2]

Meanwhile, due to the emergence of a new class of applications which emphasized on the portable electronic devices, there was a demand to develop ultra-low power ICs, considering power dissipation to be the primary design criterion. Further, with progress in scaling of devices, it is possible to integrate a large number of transistors on a single die with smaller feature lengths and supply voltages. This induces significant economic and technical difficul-

ties on the yield because of increased on-chip power dissipation. By the early 1990, it became clear that there was a growing need of high performance circuits in the semiconductor industry to support the market demand. This directed the design process in 1990's to focus on developing a number of efficient design approaches for optimizing speed and power of circuits. A series of several lines of commercial microprocessors considering the evolution of power consumption is shown in Figure 1.2. Furthermore, increased circuit complexity and aggressive device scaling have imposed several deleterious effects on the development of high speed ICs, such as signal delay uncertainty, on-chip clock jitter, noise margin degradation, etc. As avoiding such effects is no longer possible in the design of high speed ICs, it is necessary to shift the design focus towards low power. With the advent of deep-submicron technologies in twenty

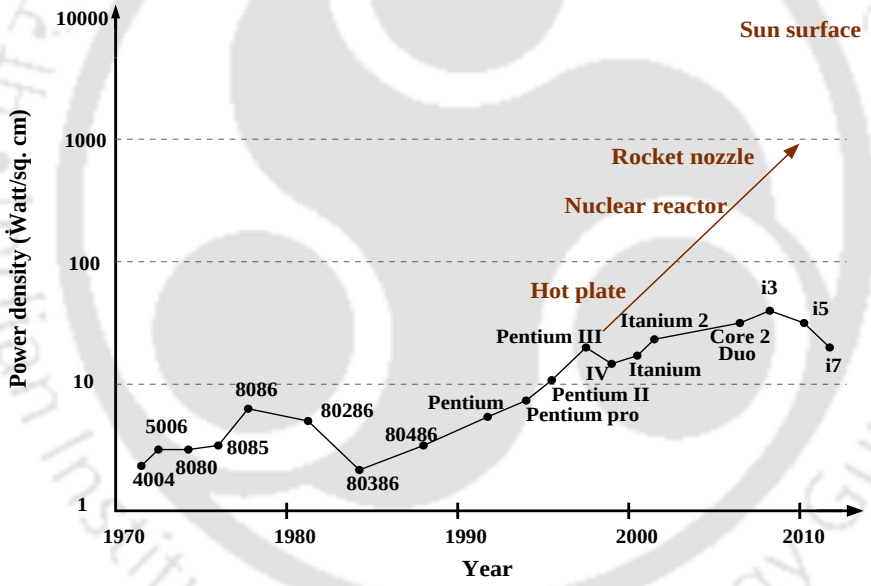


Figure 1.2: Evolution of microprocessor power consumption (Intel microprocessor family) [2].

first century, device scaling is continued to keep up with the Moore's law. Accordingly, the overall power dissipation of a chip continues to increase indeterministically . With increase in on-chip power dissipation, it is evident to come up with new challenging and expensive solutions, such as liquid cooling technologies, which significantly increase the overall cost of the system. Moreover, with explosive growth of portable and handheld devices, mobiles, PDAs, tablets, etc., a technical solution is needed to account for the tradeoff among silicon

area and on-chip power dissipation. Since the emphasis is on to develop ultra-low power high performance ICs, multi-core systems were developed in the first decade of twenty first century and the design effort was focused on reducing power dissipation of high performance chips without affecting the reliability. These changes are reflected in the growing timeline of speed, power and noise as shown in Figure 1.1.

1.2 Problem of power distribution and analysis

Power distribution networks help in distributing the power across different functional blocks of the chip. To maintain the signal integrity across the integrated circuit, it is crucial to analyze voltage drops in the metal lines carrying power (V_{dd} and V_{ss}) to different blocks. These metal lines form the power distribution network of the chip. In the current scenario, V_{dd} and V_{ss} have scaled down to 0.9V and 0V, respectively, to reduce power consumption of the circuit. Because of the parasitic resistance offered by these power metal lines, a small drop in voltage (for example 10-15mV) is sufficient to propagate incorrect logic inside the chip. Thus, it has become critical to find out potential distribution of these metal lines forming power distribution network so that adequate precautions can be taken while designing the circuit. This is achieved by analyzing power distribution network of the electrical circuit being designed.

As mentioned in section 1.1, with higher chip density, a number of secondary effects have also become of vital importance during IC design process. These effects make the circuit designer's job very critical by imposing a number of constraints while designing a power distribution network. Under shorter transition times, smaller noise margins and increased current densities along with smaller feature lengths and parameters, variabilities across the integrated circuit have become more significant. Such power supply variations affect the functionality of a chip by introducing noise in the form of voltage drops and chip leakage currents.

The problem of power delivery is illustrated with an example in Figure 1.3, where a basic power delivery system is shown. The system consists of a 1V power supply, two loads having

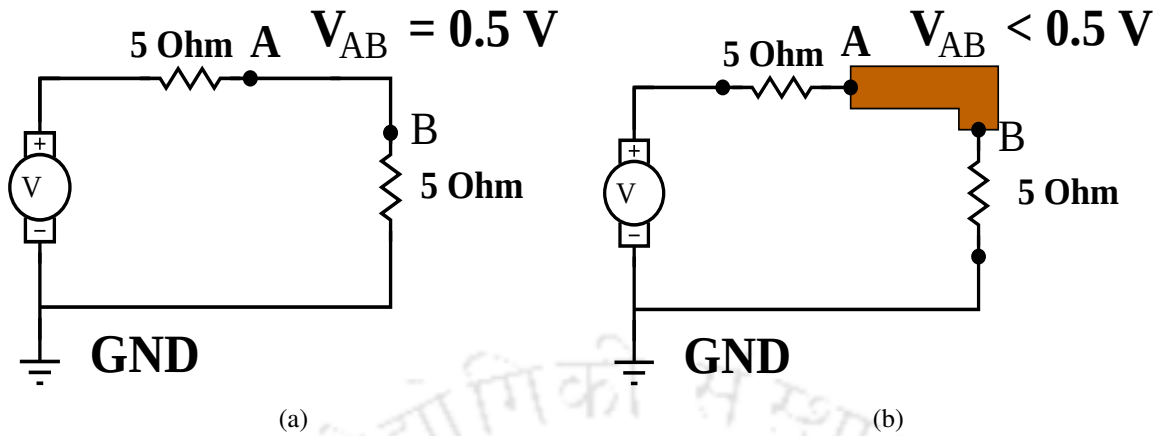


Figure 1.3: A basic power distribution system, (a) with no metal resistance (ideal condition), (b) with metal resistance.

resistances 5Ω and interconnect lines connecting the supply to the loads. It can be observed from Figure 1.3(b) that the potential, V_{AB} is evaluated to be less than 0.5V as the interconnect line between node A and node B is not ideal, i.e., the metal line has a finite parasitic resistance, $R_p = \rho \times \frac{l}{a}$. Because of resistive voltage drop between node A and node B, the voltage level of V_{AB} changes from the nominal level provided by the power supply, dropping below 0.5V between nodes A and B. This change in supply voltage is referred to as power supply noise. Power supply noise adversely affects the functionality of a chip by introducing several mechanisms. One of the mechanisms is the occurrence of worst-case voltage fluctuations (hotspots or locations having IR drop) across power distribution network of a chip. These fluctuations become more significant with increase in size of power distribution networks. Thus, the primary objective is to design a power distribution system which can supply sufficient current to each transistor on an integrated circuit while ensuring that the power noise does not exceed certain range near the nominal voltage levels (target noise margins) [2].

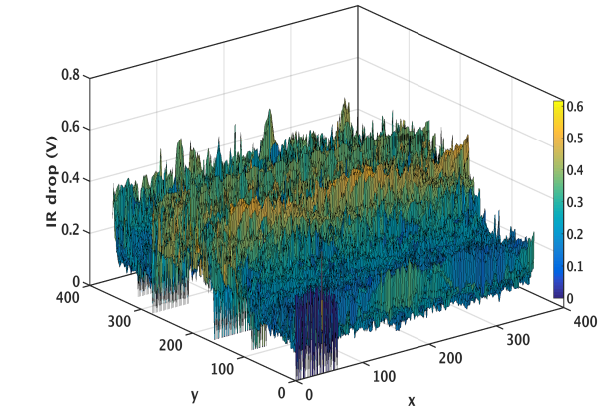
1.3 Problem of power distribution network design

With decrease in feature size and supply voltage, ensuring proper functionality of an integrated circuit (chip) is imperative. The decrease in supply voltage brings out significant

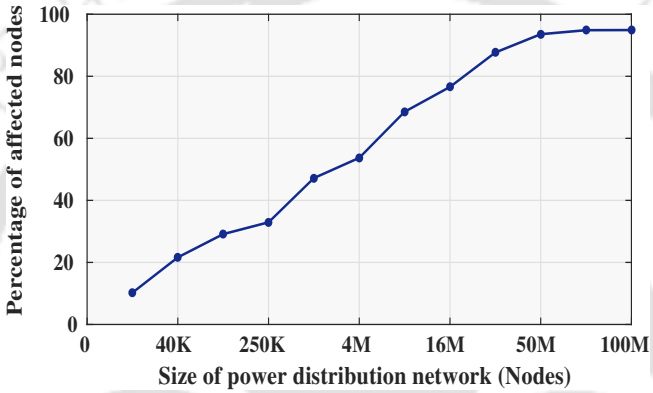
challenges into the design of power distribution networks [3]. With lower supply voltage, issues related to supply voltage fluctuations (hotspots) can not be ignored. These issues can severely affect the functionality of an integrated circuit by introducing signal irregularity and logical faults.

For example, an integrated circuit with die area $7mm \times 7mm$, line width of $50\ \mu m$, sheet resistance of $0.5\ m\Omega/mm^2$ and distributed power distribution pins on the periphery of the chip creates a voltage drop of $35\ \mu V$ on a single node placed at center of the power distribution network, if $1\ mA$ current flows through the power distribution network. Further, with increase in size of power distribution network, the voltage contours typically degrade as the signal (current) travels from periphery to center point of the chip as shown in Figure 1.4(a). On the other hand, according to the technology projections from International Technology Roadmap for Semiconductors (ITRS) [4], with progress in technology, average current entering the integrated circuit increases substantially over time. This increased flow of current allows increase in gate-switching speeds, which imply the occurrence of hotspots (large voltage drop or IR drop), $L(di/dt)$ noise [5] and electromigration phenomenon [6]. Consequently, with increase in size of the power distribution network, the network is more susceptible to several adverse effects. It can be observed from Figure 1.4(b) that with increase in the size of the power distribution network, there is a nonlinear increase in the percentage of nodes affected by IR drop. This enforces the designer's attention towards careful power distribution network design. A robust design of power distribution network involves repeated planning and careful refinement using suitable optimization methods. Such optimization methods need to be equipped with efficient search strategies to administer multiple variables with high computational efficiency.

Further, the decrease in supply voltage allows a smaller margin of voltage drop across the power rails during power distribution. As power distribution network consumes a large portion of routing resources (almost 30% or more (below 28nm)), with increase in the size of power distribution network and supply voltage variations, the voltage drops across these power rails (due to metal resistance) cannot be pushed aside [4]. Therefore, the power dis-



(a)



(b)

Figure 1.4: (a) IR drop distribution across $ibmpg2$ benchmark with maximum voltage drop (IR drop) at the center region (yellow region), (b) Percentage of affected nodes with increase in size of power distribution network.

tribution network needs to be designed with a wider width than signal nets and minimizing the power network area can reduce the routable space and save it for system performance improvement. Moreover, the magnitude of IR drop and variations depend on the design of a power distribution network, mostly on wire width. As wire width accounts for the power network area, minimizing IR drop also affects the area of the power distribution network. Furthermore, as the amount of IR drop at a particular position substantially affects the nearby region, it is necessary to minimize the IR drop of the entire power distribution network by considering each hotspot region individually subject to area, effective width and current constraints. Detailed formulations of both IR drop and area minimization problems are discussed in chapter 4.

1.4 Problem of analog circuit sizing

As silicon process geometry is reduced in scale, the underlying circuitry on a chip has become more complex. The cost of designing such complex systems drives towards the path of employing automation in the field of analog circuit design. Because of higher levels of integration inside a chip, there is a need to optimize dozens of performance specifications, such as gain, power, unity gain bandwidth (UGW) etc., with a whole new set of problems across different levels of design abstraction. Therefore, a particular circuit topology can be exploited by any efficient design technique to search for a viable solution that may be close to optimum. Depending upon the number of design objectives, the problem of circuit sizing can be classified into two categories., i.e., single objective optimization problem and multiobjective optimization problem. A brief overview on the two categories of the problem of circuit sizing is given below.

1.4.1 Circuit sizing as a single objective optimization problem

Any analog circuit design problem can be transformed into its convex form with circuit response being formulated as an objective function subject to a variety of design constraints, i.e. circuit specifications, such as DC gain, power consumption, noise figure (NF), phase margin (PM), gain-bandwidth product (GBW), input common mode range (ICMR), etc. If the circuit response and design specifications are mapped to a unique cost function, then objective function can be bounded by several design constraints. An example of the problem of maximizing DC gain of a two-stage operation amplifier circuit is given below,

$$\begin{aligned} &\text{maximize } \texttt{DC gain} \\ &\text{subject to DC gain} \geq 55\text{dB}, \qquad \text{power} \leq 0.3\text{W}, \\ &\qquad\qquad \text{PM} \geq 60^{\circ}, \qquad \text{GBW} \leq 30\text{MHz}, \\ &\qquad\qquad 0.8 \leq \text{ICMR} \leq 1.6. \end{aligned} \tag{1.1}$$

1.4.2 Circuit sizing as a multiobjective optimization problem

Circuit sizing requires the knowledge of design objectives (design specifications as function of design variables) to be minimized or maximized and the knowledge of competing design constraints (tradeoffs among competing design specifications). As the main purpose of circuit sizing is to explore the tradeoffs exhibited by design specifications, circuit sizing problems need to be expressed as multiobjective constrained optimization problems. In multiobjective constrained circuit optimization, design specifications ($i.e.$, power consumption, DC gain, slew rate, GBW, phase margin, etc.) are considered as objectives or constraints. While formulating a multiobjective constrained optimization problem, specific design specifications are formulated as objectives and others are considered as constraints. As knowledge of tradeoffs among all design specifications is quite optional to the designer, only specific design specifications are considered as objectives during optimization. To avoid any erroneous performance in analyzing tradeoffs during design, both severe and flexible design constraints, such as limiting the increase or decrease transistor sizes, limiting the operation region of transistors, etc., are defined. Mathematically, an example of multiobjective constrained circuit optimization problem in analyzing an analog amplifier circuit[1] can be expressed as,

$$\begin{aligned} &\text{minimize } \texttt{power} \\ &\text{maximize } \texttt{DC gain} \\ &\text{subject to DC gain} \geq 55\text{dB}, \qquad \text{power} \leq 0.3\text{W}, \\ &\qquad\qquad \text{PM} \geq 60^{\circ}, \qquad \text{GBW} \leq 30\text{MHz}, \\ &\qquad\qquad 0.8 \leq \text{ICMR} \leq 1.6, \end{aligned} \tag{1.2}$$

where PM, GBW and ICMR denote phase margin, gain-bandwidth product and input common mode range of the amplifier circuit, respectively. As most of the multiobjective optimization techniques are developed as minimization techniques, the objective that is to be maximized is converted to a minimization problem by inverting its sign [7].

[1] Details on amplifier circuit is described in chapter 5.

1.5 Problem of signal integrity in memristor crossbar array

Recent advances in the realization of memristor devices have opened unique ways to deal with future technology starting from manufacturing of nano-devices to neuromorphic computing. With an apprehension of approaching limit upon scaling down CMOS devices in near future, it is necessary to have a pervasive study on emerging technology, such as quantum computing [8], spintronic [9] and memristive devices [10]. In this regard, the application of memristor crossbar array in the realization of neuro-biological architecture is observed to be promising [11]. The memory wall problem of Von Neumann architecture [12] can be pursued to great extent by attributing to such non-volatile memory crossbar arrays having low-power consumption and fast switching characteristics.

However, with the increase in density of integration of memristors and size of crossbar array, various challenges rejuvenate themselves, such as issues related to increase in sneak path currents, high parametric variation during device operation, IR drop along the memristive network [11], etc. One of the significant issues among these challenges is the IR drop across memristor crossbar arrays (MCAs). There are many factors which add to IR drop across MCA. The first set of factors arises due to high-density designs using memristors having smaller feature sizes inside MCAs, which operate at lower supply voltages [13]. At these voltages, instances of large IR drops across the memristance path become significant, which severely degrade the performance of MCA. In addition, with the rise in demand to realize complex computing architectures, there is a need to design increasingly complex and large MCAs at lower supply voltages. This enhances the chances of IR drops across the crossbar array as the wire resistance and power consumption increases with increase in size of MCA. Several techniques have been presented in the literature for IR drop compensation across MCAs, which are used to realize neuromorphic computing systems (NCS). These techniques attempt to minimize the IR drop problem across MCA by transforming it into a linear programming problem (LPP). The objective of solving this minimization problem is to reduce the read/write error at the output by training and assigning arbitrary weights along the wires [13].

Several factors, such as leakage current, IR drop threshold, metal width, etc. impose a significant impact on the total IR drop across MCA. The natural response for balancing these factors to limit the amount of IR drop is more conservative during the design of MCA.

1.6 Need of efficient EDA tools

Contemporary design and manufacturing of large and complicated integrated circuits require different forms of circuit analysis performed using suitable computing environment. With rapid changes in design styles, circuit analysis is not limited to prediction of circuit responses, rather to verify the complex design styles and circuit performance. This gives rise to dynamically developing areas, such as EDA and CAM to bridge the gap between microelectronics and computer science. The EDA tools presently available comprise of various design frameworks, which encompass logic and circuit simulators, a layout editor, etc. Similar, though powerful tools [14–16] are being employed in VLSI design industry, which are developed and commercialized over the years by keeping the ubiquitous SPICE simulation program [17] as an integral part of these EDA software stack.

Apart from circuit analysis, EDA tools are also used in conjunction with optimization programs or techniques to achieve designs which optimize certain aspects of circuit quality and it is accomplished by optimizing (minimizing) deterministic and/or statistical objective functions which represent various design specifications or parameters. With changes in levels of circuit complexity and design abstraction, it is necessary to employ numerical optimization techniques that make use of sizing principle and biasing performances of all devices in the circuit to meet certain design constraints. Once an appropriate circuit topology is selected during design process, certain performance specifications are represented as objective functions by formulating a set of physical equations that relate device characteristics to design parameters. Further, due to the need to analyze the tradeoffs among various design specifications in the presence of process variability during circuit design, it is necessary to observe the variations in performance specifications which can be mapped into a multi-dimensional

objective space (Pareto space), spanned by competing objective functions (performance specifications) produced under design constraints. Analyzing multiple objective functions (two design specifications) simultaneously requires complete exploration of design space, and increase in number of design variables accelerates the expansion of this space [18]. Further, with addition of design constraints to the design metric, the design optimization problem becomes complex, i.e., presence of multiple local optima, increase in dimension and epistasis (parameter interaction), etc. In such cases, it is difficult to solve the design optimization problem by exact optimization methods, and approximate algorithms (metaheuristics) can be opted as an alternative approach.

1.7 Metaheuristics

Although many "classical" algorithms (both iterative algorithms and geometry based algorithms) are presented in literature with several modifications for both analysis and optimization of circuits, performance of these classical algorithms become ineffective with increase in size and complexity of the problems. However, the advent of metaheuristics can be marked as a reconciliation of both circuit analysis and circuit design optimization domains. Further, as circuit modeling and analysis are often affected by uncertainties and errors at many levels, it is necessary to develop suitable techniques to avoid the presence of inconsistent solutions during analysis and to handle problems which comprise of both discrete and continuous variables. In view of this, metaheuristics can be employed to solve these problems as they can adapt to the continuous problems while solving a problem having discrete variables.

Metaheuristics are regarded as upper level search methods that can be employed as guiding strategies in underlying heuristics to solve specific optimization problems (i.e., problems with high dimension and multiple local optima). Metaheuristics are general-purpose algorithms that can be applied to solve almost any optimization problem. However, metaheuristics provide acceptable solutions in a reasonable amount of computational time, which may not be optimal. Further, the design optimization problem becomes complex with the presence

of nonlinearity in objective function or design constraints. In such situations, metaheuristics are good candidates for this class of design optimization problems as traditional approaches (exact optimization methods) may fail to provide an appropriate solution. Moreover, metaheuristics have the following advantages:

a. To some extent, metaheuristics show stochastic behavior and this allows to counter combinatorial explosion of the possibilities during analysis.

b. Metaheuristics have an inert advantage of not resorting to problematic calculations of the gradients of objective functions and they lend themselves to all kinds of extensions.

1.8 Motivations and contributions of the thesis

Based on the problems defined in previous sections and application of metaheuristics in circuit analysis and optimization, the contributions of the thesis and motivation of the proposed work are presented in this section.

- **Power distribution network analysis using metaheuristics**
  - One of the major concerns in today's CMOS VLSI design is reliable on-chip power delivery. As semiconductor technology continues to scale down day-by-day, different process variabilities in silicon keep on manifest themselves affecting the chip performance. One of the critical process induced variations come from worst-case voltage fluctuations (hotspots) across power rails of a chip. These fluctuations have become more significant with increase in size of power distribution networks. Thus, it is necessary to locate the hotspots accurately throughout the power distribution network for efficient design verification by using suitable computing environment and a methodology.
  - The objective is to design a parallel and scalable power distribution network analysis tool, which can locate the hotspots across the power distribution network accurately. Further, the analysis tool must have the ability to perform both steady

state and transient analyses on very large power distribution networks (having more than 30 million nodes) using minimum number of computations and memory with acceptable accuracy loss.

  - We present two techniques based on random walk (two-step random walk (TSRW)) and river formation dynamics (RFD) methods to analyze large power distribution networks. The effectiveness of both techniques are analyzed by running several experiments on GPU. Experimental results show that both the proposed TSRW and RFD methods accelerate the analysis to achieve remarkable speedups with acceptable loss in accuracy (less than 5%).

- **Power distribution network design optimization using metaheuristics**

  - In general, power distribution network (PDN) analysis is performed as a perfunctory signoff procedure as it is employed at a later stage within the design flow to enable designers to correct major PDN-related issues, such as IR drop, $L(di/dt)$ noise, electromigration issues, etc [3,6]. As described in section 1.3, with increase in size of the power distribution network, the number of nodes affected by IR drop also increases nonlinearly. Therefore, it is necessary to minimize IR drop by redesigning the power distribution network. Further, as IR drop depends on the wire width of power rails, minimizing the IR drop also affects the area of the power distribution network.

  - Secondly, to avoid the occurrence of any major PDN problems, the designers overconstrain the designs by extending their margins and routability. Due to this overdesigning, overall power distribution area becomes too congested for a placement and routing engine to predict the amount of routing and it requires manual efforts to slice the wire area of the PDN and to complete the routing. This process of design slicing relies on designer's know-how and past design experience. However, it may result in false errors due to boundary effects and the degree of uncertainty in current directions. Therefore, minimizing the wire area of the entire PDN is

more complicated and the number of design variables can be huge with large scale PDNs.

- To minimize the wire area of the PDN, an optimization framework is presented based on river formation dynamics (RFD) heuristic. The framework is utilized to size the widths of power distribution network for very large-scale designs so that the wire area required by power rails is minimized. Both the area minimization problem and IR drop problem are transformed into two single objective optimization problems subject to various design constraints, such as electromigration constraints. The minimization process is carried out for both the problems using RFD method. The random probabilistic search strategy of RFD is employed to advance through stringent design requirements to minimize the wire area and IR drop of an over-designed power distribution network.
- Later, we modify the RFD method (MRFD) to analyze the design of several power distribution networks (minimization of IR drop and area of power distribution networks). Additional nature-inspired factors, like *transverse slope statistics* and *sediment transport rate* are incorporated to formulate a generalized model of RFD. The generalized model is proved to control flow of the water drops to ensure better convergence and stability of the MRFD method.
- Finally, experimental results are demonstrated to show performance of the proposed generalized model of RFD on both standard single objective test functions, and industry standard power distribution network benchmarks. It is observed that both RFD and MRFD frameworks show competitive performance with respect to other peer algorithms (both direct methods and metaheuristics) in optimizing power distribution network benchmarks.

• **Analog/RF circuit design optimization using metaheuristics**

- Circuit design process is characterized by predetermined specifications and care-

ful selection of design parameters. As integrated circuit technology scales down to nanometer regime, designers need to embrace new EDA paradigms while designing a robust, complex analog/RF system. For analyzing tradeoffs among various design specifications in the presence of process variability during circuit design, it is necessary to observe variations in the performance specifications which can be mapped into a multi-dimensional objective space (Pareto space), spanned by competing objective functions (performance specifications) produced under design constraints. Analyzing multiple objective functions (two design specifications) simultaneously requires complete exploration of design space, and increase in number of design variables increases size of this space. Further, with addition of design constraints to the design metric, the analysis becomes cumbersome. Therefore, it is necessary to have an efficient design methodology to capture the performance of circuit followed by an efficient constraint handling strategy.

– Our objective is to develop an efficient optimization framework for analog/RF circuit design. The optimization framework should be able to handle multiple design specifications, a number of design parameters and several design constraints, simultaneously to analyze tradeoffs among circuit parameters.

– Here, we present the application of both single objective and multiobjective optimization frameworks based on MRFD method (MRFD and MOMRFD, respectively) to design several analog/RF circuits. The single objective framework based on MRFD method discussed in previous section is extended to the application of optimizing analog/RF circuits. Apart from RFD based optimizer, a hierarchical mutation based multiobjective genetic algorithm (hNSGA-II) and an improved multiobjective framework based on hybrid of brain storm optimization algorithm and RFD scheme (IMBSO) are also developed to optimize various analog/RF circuits. The proposed algorithms are employed to design various performance specifications of a two-stage operational amplifier, a folded cascode amplifier and a low

noise amplifier circuit, and it is observed that both MOMRFD and IMBSO demonstrate competitive solutions (in terms of convergence and diversity) in achieving suitable Pareto optimal solutions for different circuit designs.

- **IR drop minimization in memristive crossbar array (MCA)**
  - With increase in density of integration of memristors and size of crossbar array, IR drop along the memristive network also increases. As we know that there are many factors that contribute towards IR drop across MCAs. The first set of factors arises from technology scaling, increased circuit complexity and high-density designs. Because of smaller feature sizes (up to micro and nanometer) inside MCAs, instances of large IR drops across the crossbar become significant at lower supply voltages. The natural response to minimize IR drop is to locate the affected regions across the crossbar network and to redesign the entire network through planning and careful refinement using suitable optimization techniques.
  - Here, we model the issue of IR drop as a single objective minimization problem subject to constraints representing various factors and analyze the optimization problem using MRFD method. To demonstrate the improvement in reduction of IR drop across MCA, experiments have been performed on MCA benchmarks for efficient evaluation of optimal IR drop. It is observed that MRFD shows competitive performance among the metaheuristics in mitigating the reliability issues in MCAs.

1.9 Organization of the thesis

To address the issues mentioned in the previous section, this thesis work is organized into seven chapters. The content of each chapter is summarized as follows:

- **Chapter 2:** In this chapter, a review on analysis and optimization of power distribution networks is presented, followed by discussion on optimization techniques used to

optimize various analog/RF amplifier circuits. Further, a preliminary review on optimization techniques to minimize the IR drop in memristive crossbar arrays is also presented. The strengths and weaknesses of different optimization techniques are discussed. A background on proposed optimization techniques is also provided. Finally, a summary of this review is presented.

- **Chapter 3:** In this chapter, two different metaheuristics based on random walk method and river formation dynamics (RFD) method are presented to analyze large scale power distribution networks on GPU. The speedup and accuracy of the proposed works are compared with different solution techniques in order to demonstrate efficiency and usability of proposed methods.

- **Chapter 4:** In this chapter, various challenges in designing a large power distribution network are outlined. A single objective optimization framework is developed based on RFD method to design large scale power distribution networks. Further, the single objective framework is modified by incorporating various factors to RFD scheme. The performance improvement in the proposed single objective framework is demonstrated by minimizing IR drop and area for large scale power distribution networks.

- **Chapter 5:** In this chapter, single objective optimization of various analog/RF circuits is presented using RFD and MRFD. Further, to analyze tradeoff among various design specifications, a multiobjective framework based on RFD is also presented, i.e., MOMRFD. Apart from MOMRFD, a hierarchical mutation based multiobjective genetic algorithm (hNSGA-II) and an improved multiobjective framework based on hybrid of brain storm optimization algorithm and RFD scheme (IMBSO) are also presented to design a two-stage operational amplifier, a folded cascode amplifier and a low noise amplifier.

- **Chapter 6:** In this chapter, optimization of IR drop across memristor crossbar arrays (MCAs) is discussed. The issue of IR drop across MCA is translated into a single

objective minimization problem subjected to constraints representing factors and the optimization problem is solved using MRFD method. The results are compared with other peer algorithms to showcase effectiveness of MRFD method.

- **Chapter 7:** In this chapter, a summary of the work presented in the thesis is discussed. Finally, a highlight to the main contributions of the work and directions for future research are discussed.

2

Methods for Performance Analysis and Design Optimization of VLSI Circuits : A Review

Contents

This chapter presents a review regarding the research work being carried out in problems from four different domains that we have discussed in chapter 1. In section 2.1, a brief review on different methodologies being proposed to perform power distribution network analysis (mostly IR drop analysis) are discussed. This is followed by an overview of different optimization techniques used to design a power distribution network in section 2.2. As the scientific community has proposed many techniques for the automation of the analog/RF circuit sizing, those approaches are briefly surveyed in section 2.3 focusing on the optimization techniques. Section 2.4 outlines the optimization methods presented to minimize IR drop issue in MCA.

With the availability of computing resources, there has been a significant growth of interest in metaheuristic domain. As metaheuristics provide acceptable solutions in a reasonable amount of time for solving complex problems in science and engineering, the usage of the family of such approximate optimization techniques has gained a lot of popularity in the past two decades. Among such optimization techniques, few algorithms are found to be most promising and successful techniques (peer algorithms), such as genetic algorithm (GA), differential evolution (DE), particle swarm optimization (PSO), etc. Section 2.5 outlines these peer algorithms and their variants. In section 2.6, the working aspect of a few metaheuristics are discussed. These metaheuristics are used to design variants of metaheuristics in the proposed research work for solving the problems in four different domains. Finally, an important aspect common to all the metaheuristics, i.e., performance evaluation, is presented along with a brief description on basic answer to the selection of any metaheuristic for a certain problem.

2.1 VLSI power distribution network analysis

Many algorithmic techniques have been proposed in the past to address issues related to reliability and design space of power distribution networks. These proposed methods tend to incorporate direct solvers, iterative solvers and heuristics based methods to perform very large industry-scale power distribution network analysis. The use of direct solvers, such as LU de-

composition method [19], to perform power distribution network analysis is well-defined in literature. However, with the available of state-of-the-art computing resources, these direct solvers do not exhibit efficient performance in terms of speed and memory usage while analyzing large power distribution networks. In comparison, iterative solvers, such as PCG, GMRES [20], etc. are more efficient in memory management, but their optimal convergence depends upon the choice of preconditioners [20]. Methods based on hierarchical approach [21], domain decomposition [19] and grid reduction techniques using multigrid [22] have been proposed to analyze the power distribution network by extrapolating the results to original grid network. Improvements to these methods have been proposed to handle issues related to accuracy [23] and memory. In addition, for carrying out efficient simulations on high performance computing platforms, many parallel strategies [20], [24], are also proposed along with a number of partitioning schemes [21] to address memory and convergence related problems efficiently by exploiting the structure of power distribution networks. Although these techniques perform well on power distribution analysis, each time the entire power distribution network is reanalyzed even though the power grid is locally modified. In such cases, various approaches are presented in literature [25], [5] to take the advantage of known information of a power distribution network to simulate and verify the updated power distribution network. However, new methods need to be addressed in order to exploit the architecture of very large power distribution networks more prominently for efficient realization.

2.2 Power distribution network design

As it is necessary to have a pertinent design of power distribution network, minimizing the issues related to voltage drop (IR drop) can save space for performance improvement. Many research works have been presented in literature to design power distribution network through minimizing different parameters, such as metal area, decoupling capacitance and voltage drop [26–30]. One of the solutions to large IR drop is to use wider wire than signal nets or to add minimum decoupling capacitors keeping reliability, area and current consumed

by power distribution network as constraints [27, 31, 32]. As power distribution network consumes a significant portion of routing resources, it is also crucial to minimize the power distribution network area using suitable optimization techniques [31, 32]. Many approaches have been proposed to minimize wire area by constructing a mesh or tree topology of power distribution network [33]. Further, these approaches [32, 33] suffer from poor convergence [34] or increased space/time complexity with increase in size of power distribution networks. Many research methodologies have also been presented to optimize the number and location of power supply PADs, in order to constrain the amount of IR drop [35, 36]. However, these research works do not emphasize the optimization of the entire power distribution network.

2.3 Optimization of analog/RF circuits

There has been continuous efforts over the past years to develop new efficient design techniques for finding solutions to design optimization problems [37]. In the prior solutions of constrained-aware design optimization (single objective optimization) [38, 39], the key challenge is to mitigate the computational cost while capturing prominent functionalities of circuits using a class of macromodels. However, all these techniques require sampling of each model that led to a compromise in accuracy while using less number of samples (Monte-Carlo samples). The cost of estimating the performance specification of the circuit increases with number of samples used during evaluation. Therefore, new approaches [40, 41] are presented in literature to handle the design complexity of integrated circuits. These approaches appear to translate the problem of circuit design into a function minimization or maximization problem, which can be solved through numerical techniques [40].

Mostly, it starts with the development of an efficient performance evaluation framework, which evaluates the minimization (or maximization) function [40]. This function is created based on the behavior of circuit topology [37, 40, 42] and they rely on the analytical formulation. The efficiency of the evaluator lies in the efficient design of circuit equations. Although a number of deterministic algorithms have been presented to perform such types of circuit

design, the uncertainty lies in the assessment of initial design point, which requires prior knowledge or additional design equation formulation. Further, with presence of uncertainties at many levels because of the complex circuit design process, analyzing a single cost function (single objective optimization) leads to faults and inconsistent solutions. This motivates the use of multiobjective optimization techniques during circuit design to explore the multi-dimensional design space (Pareto space) by correlating the performances of multiple design specifications (Pareto fronts).

Several multiobjective optimization techniques are presented in literature to handle the uncertainty during circuit design. Various multiobjective evolutionary algorithms, such as NSGA-II [7, 43], MOEAD-DE [44], MOPSO [45], etc. are used in various analog and RF circuit synthesis tools. Besides these research outcomes, several commercial design tools are also developed for circuit analysis and design in the past few years, $i.e.$, Barcelona Design [14], Genius's ADA [15], NeoCircuit [16], etc. However, in aforementioned techniques, extracting the Pareto fronts among multiple design specifications subject to various design constraints becomes a challenging task as it requires complete exploration of design space in each evaluation to maintain design accuracy.

2.4 IR drop minimization of memristor crossbar arrays

As the scaling of conventional CMOS devices is approaching the limit, scalable nano-devices, i.e., resistive (memristor) [10, 46], nanotube [47, 48], sprintronic [9, 49], etc. have emerged as promising alternative. It is anticipated that highly parallel computing architectures equipped with these devices would exhibit greater CPU performance and memory bandwidth. This motivates many researchers to come up with highly parallel algorithms to realize such complex computing architectures. Neuro-biological architecture is found to be one of the promising candidates, which provides a neuromorphic environment to realize various complex computing architectures using nanoscale resistive devices, e.g., memristors.

The programmable resistance state of memristors is adapted to realize the changing states

of biological synapses. In response to the need of high integration density and low power, a crossbar structure consisting of memristors (i.e., MCA) is developed [11, 50] and is regarded as a suitable option to improve the execution efficiency of Matrix-Vector multiplications [11]. However, as mentioned in previous chapter, implementations of such neuromorphic structure using memristor has come with several technical challenges. Besides physical limitations of the hardware circuit [11], the scale of single MCA is limited by the occurrence of IR drop along crossbar network consisting of metal wires and memristors. It is observed that there is severe voltage degradation towards far end of the crossbar array due to IR drop on a 64×64 crossbar [11]. With further increase in the size of MCA, the impact of IR drop also becomes critical, resulting in logical or functional failures of the entire system. As IR drop affects the training and sensing operations of MCAs, one of the strategies is to reduce the weight matrix to minimize the computation in the crossbar [13]. The outcome of this approach results in decreased size of crossbar and improved computational reliability of the neuromorphic computing environment. However, the process of system reduction and IR drop compensation scheme presented in [13] makes it a complex procedure to approach from point of view of overall implementation and computational cost. This requires the user to have prior knowledge of training database and with increase in size of the MCA beyond 128×128, it becomes inefficient to realize the weight matrices associated with the network. This supports the limitations of this approach and speaks for a research challenge to adopt new optimization techniques to minimize IR drop across MCA.

2.5 Peer metaheuristics

The use of metaheuristics to solve real world problems is widely accepted within the research community. These metaheuristics provide high quality solutions to important problems in business, engineering, economics and science in reasonable amount of time. Although finding exact solutions in these applications is still a real challenge despite recent advances in computer technology, metaheuristics seem to be the methods of choice in many decision

making processes. These decision making processes are increasingly complex and more compute intensive due to more decision variables being used to model complex systems and more input data or parameters being utilized to capture the complexity of problem instances. In such cases, many metaheuristics have been proposed in literature and few of them are widely accepted as state-of-the-art methods, called peer metaheuristics. In this section, we give an insight into such peer metaheuristics, e.g., genetic algorithm (GA), differential evolution (DE), particle swarm optimization (PSO), etc.

2.5.1 Genetic algorithm (GA)

Being introduced in 1975 by Holland [51], genetic algorithm (GA) has emerged as a practical, robust optimization and search method over the past few years. GA is a population based search algorithm that simulates natural evolution. The search space of GA is characterized by a collection of population individuals that puts a great deal of emphasis on the combined interactions of selection, recombination, mutation and crossover operations acting on such individuals. The objective of natural evolution in genetic algorithm is to find the individual from the search space with the best genetic base (i.e., the chromosomes with the best chance of survival). An overview of working principle of genetic algorithm is described in Algorithm 1. GA starts with an initial population generation. Then, the quality of the individuals in the population is determined and few individuals are chosen as parent population. The quality of the individual is measured with an evaluation function. A child population is generated using recombination or mutation and crossover operations from parent population. Further, a few individuals are removed from the population according to the selection criterion in order to reduce the total population to the initial size. The process is continued for a number of iterations, which are referred as generations. The individuals having better quality survive through the generations and represent the natural evolution process. There are several forms of GAs that use different mutation and crossover operators [52, 53] to increase the probability that the algorithm results in near-optimal solution in a reasonable number of iterations. Mutation is needed to explore new individual instances and helps the algorithm to avoid local optima.

Algorithm 1: Genetic algorithm (GA)

```
Generate initial population at random
while (not stop ) do
  Select parents from population
  Produce child population from selected parents
  Mutate individuals
  Merge the child population with the main population
  Reduce the population by selection
end while
Output the best individual found
```

Crossover operation helps in increasing the average quality of the population. Performance of these operators can define the quality of fitness function being evaluated in each generation. As some sort of heuristic measure, this fitness function defines a measure of profit or quality of the solution for the underlying problem. Mostly, GA has been employed as a stochastic procedure to obtain global optimum solution for different combinatorial optimization problems such as scheduling problems [54], traveling salesman problem [55, 56] and in machine learning [57]. Further, GA has been applied to solve both single objective, multi-objective and many objective problems in literature [7, 58].

2.5.2 Differential evolution (DE)

Differential evolution (DE) [59] is one of the most popular evolutionary algorithms available in literature. The algorithm functions by generating new candidate solutions in each generation that are created by combining the parent individual and several other individuals of the same population. If the newly generated candidate has better fitness, it replaces the parent individual in the next generation. The new candidate solutions are generated using $DE/current\text{-}to\text{-}best/1$ mutation scheme followed by $rand/1/exp$ crossover scheme in each generation. The mutation scale factor (F) and the crossover probability (C_r) are generally set to 0.8 and 0.9, respectively during implementation of DE. The setting of the two parameters, i.e., scale factor (F) and the crossover rate (C_r) is neither intuitive nor experimental, but rather a crucial one for the overall performance of DE algorithm. Several research methods

have been presented in the literature to study the settings of these two parameters and a series of analysis has been performed to recommend a stable bound for setting F and C_r [60, 61].

Algorithm 2: Differential evolution (DE)

```
Read values of the control parameters of DE: F, C_r, and
population
size N.
Generate initial population at random
while (not stop ) do
  Generate donor population corresponding to target population via
  the DE/current-to-best/1 mutation scheme of DE
  Generate a trial population for the target population using
  rand/1/exp crossover scheme
  Evaluate population
  Selection better individual from target population
end while
Output the best individual found
```

However, it has been observed over the years that an efficient parameter setting in DE is very dependent on the type of problems, which validates the No Free Lunch Theorem [62] with reference to the DE variants. Apart from this parameter variations in DE schemes, different mutation strategies have been employed to offer an alternative perspective to the DE search and tend to increase the exploitative pressure within the search space to maintain diversity [61]. There are also several variants of DE having different mutation strategies and employment of self-adaptation of parameters, such as SADE [63], JADE [64], etc. A more sophisticated and efficient variant of DE scheme is LSHADE [65, 66]. LSHADE and its variants along with SADE are chosen for an extensive comparison in following chapters of the thesis. These DE schemes are chosen because of their efficiency in solving various problems. These schemes have been the object of interest of high quality research for a long time.

2.5.3 Particle swarm optimization (PSO)

Particle swarm optimization (PSO) utilizes a search procedure in which a swarm of particles are allowed to move in a search space with random velocity. Fitness values are evaluated

for each movement of particle. Depending upon the fitness values, the personal best position (x_{pbest}) of the particle and global best position (x_{gbest}) among the swarm of particles are evaluated in each iteration. According to the updated personal and global best positions, the position (x) and velocity (v) of each particle are updated using (2.1) and (2.2), respectively [67].

$$x_{i+1} = x_i + v_{i+1} \tag{2.1}$$

$$v_{i+1} = w_i v_i + c_1 r_1 (x_{pbest} - x_i) + c_2 r_2 (x_{gbest} - x_i) \tag{2.2}$$

$$w_{i+1} = (w_{max} - w_{min}) \left[\frac{T - i}{T} \right] + w_{max} \tag{2.3}$$

where, x_i and v_i denote position and velocity of a particle in i^{th} step respectively; c_1 and c_2 represent acceleration coefficients; r_1 and r_2 denote random numbers within range [0,1]; w represents inertia used for velocity calculation; w_{min} and w_{max} denote the minimum and maximum values of inertia w, and T denotes maximum number of iterations.

PSO makes use of the social association between individual particles to choose a random path in the multi-dimensional search space for achieving a global optimal solution. It relies on the efficient selection of constriction coefficients (w) to explore the feasible space of operation with a flexibility to exploit alternate search spaces for diversity preservation [68]. Due to the randomness involved in the variation of parameters in PSO, often particles tend to become trapped in local optima. Further, the lack of dynamic adjustment in velocity of particles result in the movement of entire swarm of particles toward local optima, causing premature convergence. With respect to the problem of premature convergence, several modifications are proposed, and a number of variants of PSO algorithm are reported in the literature [69] based on various aspects for single objective optimization, such as quantum-behaved PSO [70], chaotic PSO [71], fuzzy PSO [72], craziness-based PSO (CRPSO) [73, 74], hybrid PSO [75], PSO with Lévy flight [76, 77], PSO with aging leader and challengers [78], etc. Moreover, in order to solve multiobjective test problems, MOPSO [79] is presented, in which the cost function makes use of Pareto dominance while moving the swarm particles in each iteration and the nondominated solutions are stored in an archive to approximate Pareto front. Several

modifications to MOPSO, such as SMPSO [80], dMOPSO [81], etc. are also presented to improve the convergence of algorithm and diversity in solutions. A few of these proposed PSO variants are chosen for comparison with the proposed metaheuristics in the thesis.

2.6 Background on Proposed Metaheuristics

2.6.1 Random walk (RW)

Random walk is a classic stochastic process in which each state changes in a random probabilistic manner. The simplest form of random walk can be represented by giving an example of a person who starts walking from the origin in a 2D plane [1] (as shown in Figure 2.1) and takes left or right path in a random way. It is similar to tossing a weighted coin at each step and wait for a probable outcome. Based on head or tail, decision of left or right path is taken, respectively.

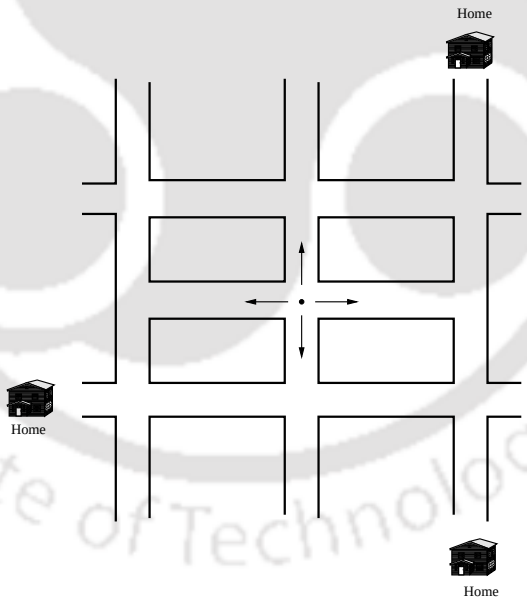


Figure 2.1: Random walk by a drunkard in a city

Suppose that a drunkard is allowed to roam in a city where the roads are laid down in a square format as shown in Figure 2.1. He can move in any direction at each cross-section with a probability of $\frac{1}{4}$ and while passing a corner or cross-section, he has to pay penalty towards his traveling expense. When the drunkard reaches at one of his destinations, he is rewarded

for reaching home and the walk comes to an end. As there are more than one destinations, the drunkard can end his walk anywhere in the city having different rewards for different walks [82]. If $p_{x,y}$ represents the probability of the drunkard going in any direction y starting from x, then

$$p_{x,1} + p_{x,2} + p_{x,3} + ... + p_{x,k} = 1, \tag{2.4}$$

where, x is the starting point; k is the number of paths from x and y can be $1, 2, 3, ..., k$. If r_x can be represented as the amount paid at each cross-section by the drunkard until he arrives at one of the destinations, the gain function $f(x)$ can be expressed as [82],

$$f(x) = \sum_{y=1}^{k} p_{x,y} f(y) - r_x \tag{2.5}$$

Random walk is one of the most widely adopted method for the analysis of VLSI power distribution networks composed of resistances, voltage sources and current sources. This method can also be applied for the analysis of power distribution networks composed of other passive elements, such as inductors and capacitors. Earlier, many methods have also been proposed to improve the computational time and accuracy in solving a power grid network with generic random walk algorithm [82], [83]. The random walk algorithm has already shown its robustness in solving 3D power distribution networks [21]. Therefore, a parallel random walk algorithm is proposed in the thesis and it is employed to analyze large power distribution networks on GPU. The details on this proposed parallel random walk algorithm is described in chapter 3.

2.6.2 River formation dynamics (RFD)

River Formation Dynamics (RFD) is a heuristic based on the formation of rivers by water drops. Water drops transform the landscape by gradually depositing debris and wearing away soils, rocks, etc. to follow the path of decreasing altitudes. Drops probabilistically tend to choose the landscapes by using a decreasing gradient principle [84]. Alternatively, it can be said that the decision of choosing the landscape is based on composing and reinforcing

the best gradient values. Different drops follow different landscapes to reach destination. From movement of drops, gradients are calculated by evaluating down slopes when altitude changes across different paths. These gradients show direction to a number of potential paths from point of origin of water drops to destination (*sea*). One of the potential paths is chosen to be the best possible solution by analyzing the cost among followed paths. The applicability of RFD scheme has been shown in literature to analyze a number of NP-complete problems [85]. In fact, several methods have also been presented to extend RFD scheme for optimization problems, such as wireless sensor networks for optimal data aggregation tree [86]. Although a number of methods have been presented on RFD, the effectiveness of RFD needs to be demonstrated by analyzing and solving various other real-life problems.

To showcase the applicability and efficiency of RFD, it is applied to perform power distribution network analysis in the thesis. Further, the application of RFD is extended to perform minimization of area and IR drop in power distribution networks, analog/RF circuit sizing and minimization of IR drop in memristor crossbar arrays. The details of the implementation of RFD and modified RFD (MRFD) methods are described in the following chapters.

2.6.3 Nondominated sorting genetic algorithm II (NSGA-II)

Nondominated sorting genetic algorithm II (NSGA-II) is proposed in 2002 by Deb et al [7] to handle multi-objective problems including the ability to find best possible solution with acceptable convergence. Although other approaches satisfy themselves by compensating for suboptimal performance in certain cases, NSGA-II has an appealing feature of overcoming difficulties in traditional multi-objective circuit optimization.

NSGA-II employs nondominated sorting approach for fitness assignment among individual solutions to solve multi-objective optimization problems [87]. As shown in Figure 2.2, the algorithm starts with a random generation of initial population (range of design alternatives for design variables) of N numbers. The current parent population (P_t) at t^{th} generation is used to produce offspring population (Q_t) with population size N by using polynomial mutation and simulated binary crossover operator [7]. The two populations P_t and Q_t are merged

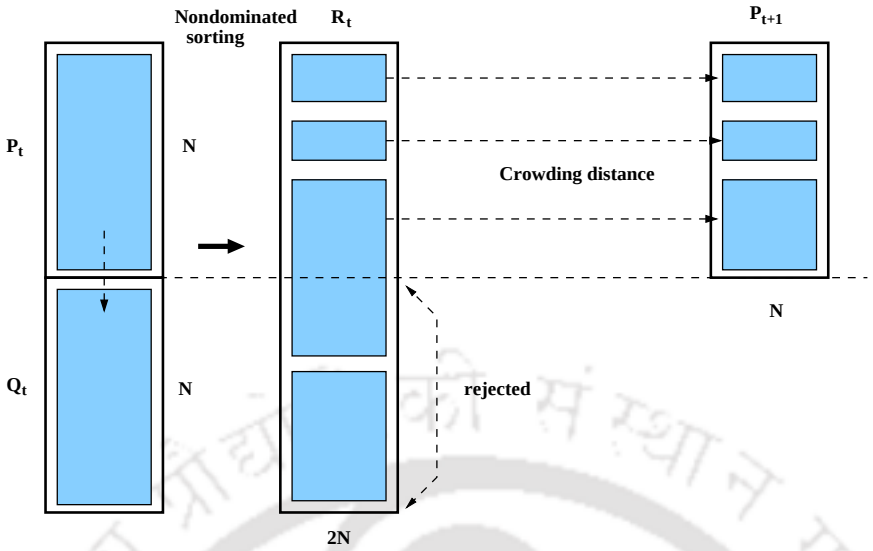


Figure 2.2: Model of NSGA-II at t^{th} generation.

together to form a mixed population R_t (of population size $2N$). The best N numbers are selected from R_t pool using nondominated sorting approach based on the dominance principle. Nondominated solutions are selected from R_t to form a population of size N in the next generation. The remaining solutions in the next generation pool are selected by employing a crowding distance scheme [7], i.e., the solutions having largest crowding distance values are chosen for the next generation to maintain a desired diversity. The application of NSGA-II has not been seriously addressed in multi-objective circuit optimization as per our knowledge. Therefore, a hierarchical version of NSGA-II (hNSGA-II) is presented in the thesis to perform analog/RF circuit optimization using hierarchy in generation of new population during polynomial mutation operation. The details of hNSGA-II is described in chapter 5 of the thesis.

2.6.4 Multiobjective brain storm optimization (MBSO)

The brain storm optimization (BSO) algorithm is evolving as a promising swarm intelligence algorithm in recent years. Since its influx as a swarm intelligence algorithm by Shi [88], various modifications are proposed to improve the performance of BSO algorithm in literature [89–91]. Further, a number of research methods are also presented in solving various

real world problems using BSO algorithm [92–94]. Due to a collective behavior of both nature-inspired pattern and data mining technique, BSO can explore and exploit the solution space naturally [88]. This collective behavior of brainstorming process employs a natural selection to evaluate an optimum solution from a cluster of solutions, where the individuals are generated by following the Osborn's original four rules [88]. However, in case of MBSO, the clustering technique is applied to explore the objective space instead of solution space [88]. MBSO uses k-means cluster algorithm as a clustering technique to group the individuals into k clusters based on each objective. The cluster having the best fitness value is selected as `Elite_cluster`. The individuals ($\mathbf{x}_i = \{x_i^1, x_i^2, \ldots, x_i^D\}$, $1 < i < N$, N denotes number of individuals and D denotes dimension) in any `Elite_cluster` are added to form an `Elite_set` (P_{elite}) and others are added to form a `Normal_set` (P_{normal}). Both `Elite_set` and `Normal_set` are temporary sets, which constitute to form an `Archive_set` ($\mathrm{P}_{archive}$), a set of all nondominated solutions. After selection of cluster(s), an individual ($\mathbf{x}_{selected}$) is selected based on cluster center(s) or random idea(s) of the cluster(s) [88]. A new individual ($\mathbf{x}_{new}$) is generated from the selected individual ($\mathbf{x}_{selected}$) using mutation operation. Both Gaussian and Cauchy mutation operators are adopted in the original proposal of MBSO algorithm to generate new population as follows [88],

$$\begin{aligned} \mathbf{x}_{new} &= \mathbf{x}_{selected} + \xi N(\mu, \sigma), \\ \texttt{or, } \mathbf{x}_{new} &= \mathbf{x}_{selected} + \xi C(\mu', \sigma'), \\ \xi &= \texttt{logsig}\left(\frac{T-t}{2K}\right) \times rand(), \end{aligned} \tag{2.6}$$

where, N is a random number generated using Gaussian distribution with mean μ and standard deviation σ; C is a random number generated using Cauchy distribution with mean μ' and standard deviation σ'; ξ is a coefficient that weight the contribution of mutation; `logsig()` denotes logarithmic sigmoid transfer function; T and t denote the maximum iteration and the current iteration numbers; K denotes a constant used for changing `logsig()` function's slope and $rand()$ function returns a random value between 0 and 1 with uniform distribution. The procedure of MBSO is described in Algorithm 3.

Algorithm 3: Procedure for multiobjective brain storm optimization (MBSO) [88]

Generate N random ideas (random population individual);
F $\leftarrow f(\mathbf{x})$; $\triangleright$ evaluate fitness for all individuals
Update $\mathrm{P}_{archive}$ according to Pareto dominance;
$\mathrm{P}_{elite} \leftarrow \phi, \mathrm{P}_{normal} \leftarrow \phi$; $\triangleright$ P_{elite}: `Set of elite individuals` , P_{normal}: `Set of normal individuals`
Cluster N random individuals into k clusters for each objective; $\triangleright$ `Assume` M `objectives`
For each objective, choose one cluster as `Elite_cluster` having the best fitness value;
if **x** $\in$ `Elite_cluster` **then**
  $\mathrm{P}_{elite} \leftarrow f(\mathbf{x}) \cup \mathrm{P}_{elite}$;
else
  $\mathrm{P}_{normal} \leftarrow f(\mathbf{x}) \cup \mathrm{P}_{normal}$;
end if
for $i \rightarrow 1$ to `max_gen` **do**
  if `random(0,1)` $<$ `p_one` **then**
    if `random(0,1)` $<$ `p_two` **then**
      if `random(0,1)` $<$ `p_three` **then**
        Randomly choose an individual from P_{elite} as $\mathbf{x}_{selected}$;
      else
        Randomly choose an individual from P_{normal} as $\mathbf{x}_{selected}$;
      end if
    else
      Randomly choose an individual from $\mathrm{P}_{archive}$ as $\mathbf{x}_{selected}$;
    end if
  else
    Randomly generate an individual as $\mathbf{x}_{selected}$;
  end if
  $\mathbf{x}_{new} \leftarrow$ `Mutation(`$\mathbf{x}_{selected}$`)`;
  $\mathbf{x}_{optimal} \leftarrow$ `Selection(`$\mathbf{x}_{selected}, \mathbf{x}_{new}$`)`;
  Update $\mathrm{P}_{archive}$ with new nondominated solutions
end for

Following the creation of new individual, MBSO algorithm evaluates the survival of individuals ($\mathbf{x}_{selected}$ and $\mathbf{x}_{new}$) based on dominance of objectives (better fitness value). If $\mathbf{x}_{selected}$ and $\mathbf{x}_{new}$ are not dominated by each other, a randomly chosen individual is selected for evaluation of new non-dominated solution. The corresponding new nondominated solution is added to `Archive_set` in each iteration and is compared with the existing nondominated solutions of `Archive_set`. The size of the `Archive_set` is controlled by truncating additional solutions using a crowding comparison operator [7].

Regardless of the diverse search movement in MBSO, there is a scope to improve the per-

formance by incorporating a number of features to the algorithmic steps. An improved version of MBSO (IMBSO) is developed and presented in chapter 5 to showcase the effectiveness in the design of analog/RF circuits.

2.7 An insight into metaheuristics performance measures

The real challenge in choosing a metaheuristic is not only to measure the algorithm efficiently in terms of quality of solution and computational cost, but also to employ quantitative or statistical analysis [95, 96] to assert the robustness of the algorithm over a wide spectrum of problem instances. This favors the usability of metaheuristics in variety of problems across diverse research communities. Further, it is highly desirable to design a general purpose metaheuristics, which can readily be employed to a wide spectrum of problem classes even with the lack of any problem specific knowledge. A number of general purpose metaheuristics have been proposed in the last fifty years, e.g., GA, DE, PSO, etc. Such metaheuristics are specifically suited for general applicability upon a large set of problems without needing any in-depth knowledge of the problem and any major in-depth modifications to the algorithm. Moreover, two or more metaheuristics can be combined to develop a hybrid methodology and can be applied to solve specific problems. Such hybrid methods are fine-tuned or tailor-made and specifically designed to analyze the mathematical properties of the problem. While such hybrid methods can provide enhanced performance over simple general purpose metaheuristics, the usability of hybrid methods reduce as an obvious consequence, i.e., the underlying concepts of such methods cannot easily be extended to a variety of problem instances.

Measuring the performance of any metaheuristic has been observed to be a complicated process over the years. The performance of a metaheuristic may vary in different circumstances according to the level of implementation, i.e., a proposed algorithm implemented by two different researchers can provide different solutions depending upon factors, such as quality of implementation skill, ability to set parameters according to problem instances, use of special data structures to fine tune the computational overhead, etc. In view of this, it becomes

arduous to compare and assess a metaheuristic and its performance with others.

According to no-free-lunch-theorem [62], no metaheuristic outperforms all other metaheuristics. In light of this well-known theorem, it might lead to an equivocal domain, whether to develop a better general framework based on a metaheuristic to effectively solve a wide spectrum of problems or, conversely, to design an application specific algorithm that fully exploits the mathematical properties of the problem. In such cases, while ranking the metaheuristics, it becomes necessary to solve the problems multiple times to have a statistical aspect. The metaheuristic which happens to solve the problem successfully in every run or most of the runs is assumed to have a better rank or performance than others for that specific problem. There has been several research studies to provide metrics for ranking different metaheuristics while solving a set of problems [96–98]. However, such metrics rank the metaheuristics by running them multiple times to gather sound statistical data, which are analyzed to assert the convergence and diversity properties of metaheuristics [97, 99].

As highlighted before, a recent line of research in metaheuristic field is concerned with the design of hybrid methods, where hybrid indicates either the combination of two or more metaheuristics [100, 101] (e.g., part of RFD stochastic search procedure has been utilized inside IMBSO framework for improved performance in chapter 5) or the intertwined usage of metaheuristics features with mathematical programming techniques [102]. As no metaheuristic can guarantee high quality solutions over all possible instances of a given problem class, a series of different benchmarks are being solved in following chapters of the thesis and their solutions are compared with other peer metaheuristics to demonstrate robust behavior of proposed metaheuristics. The solutions are compared based on the performance of proposed metaheuristics over a number of successful runs and various metrics (both quantitative and statistical) are employed to rank these metaheuristics among other state-of-the-art metaheuristics.

2.8 Summary

In this chapter, we have given a brief review on several techniques presented in the literature for analysis and optimization of problems in four different domains (i.e., power distribution network analysis, optimization of power distribution network, optimization of analog/RF circuits, minimization of IR drop in MCA). It is observed that there are not enough metaheuristics available to address the problems in the above mentioned domains. Later, an insight into several peer metaheuristics has been presented and their working principles are discussed in brief. A background on the proposed metaheuristics are also discussed and the shortfalls of traditional metaheuristics are described. Therefore, several extensions are provided to address the shortfalls of these metaheuristics in the following chapters. Further, it is discussed that the several factors play key roles in the quality of solution when metaheuristics are employed to solve specific problems. In subsequent chapters, we describe the solutions of problems in above mentioned domains in detail using our proposed methodologies.

3

Power Distribution Network Analysis and Methods

Contents

3.1 Introduction

Efficient analysis of power distribution network (power grid) is essential for the correct functionality of a chip to different load conditions and variability of elements. Nowadays, it has become critical to analyze power distribution networks as the power supply voltage continues to drop because of technology scaling resulting in higher chip density. With the increase in demand for complex low voltage integrated circuits (ICs), issues related to power distribution network have become of vital importance. One of the important issue for a chip designer is to design a reliable power distribution network for a wide variety of workloads to minimize the fluctuations in voltage levels across different functional blocks of a chip caused by increased interconnect resistances, large sizes of networks and, increased element and operation variability in lower silicon technology nodes. Therefore, it is necessary to analyze the power distribution networks for both voltage behavior and grid safety.

Today, power distribution network analysis is typically done by simulating the entire power distribution network to evaluate voltage drop at every node. Due to large sizes of power grid networks, it has become critical to analyze the whole power distribution network on a single computer due to lack of memory resources. Several attempts have been made to address the issues related to memory using parallel computing strategies [20, 24] and model order reduction (MOR) techniques [103, 104] including a number of partitioning schemes [21, 105]. However, statistical techniques like RW [82, 83] are proved to be efficient while solving large power distribution networks. Therefore, we start the analysis of power distribution network using RW method.

3.2 Power Distribution Network Analysis

3.2.1 Network Formulation for Steady State Analysis

During steady state analysis of a power distribution network, the network is usually modeled as an electrical network composed of resistors, voltage sources and current sources as shown in Figure 3.1(a). Considering a single node x of the V_{DD} grid as shown in Figure

3.1(b) and after applying Kirchhoff's current law and Kirchhoff's voltage law at node x, the algebraic equation can be formulated as [82],

$$V_x = \sum_{y=1}^{k} \frac{g_y}{\sum_{y=1}^{k} g_y} V_y - \frac{I_x}{\sum_{y=1}^{k} g_y}, \tag{3.1}$$

where g_y is the conductance of y^{th} connected nodes; k represents the number of adjacent nodes of x; V_y is the potential of the y^{th} adjacent node; I_x is the current drawn from node x, and V_x represents the potential at node x. Steady state analysis of VLSI power grid net-

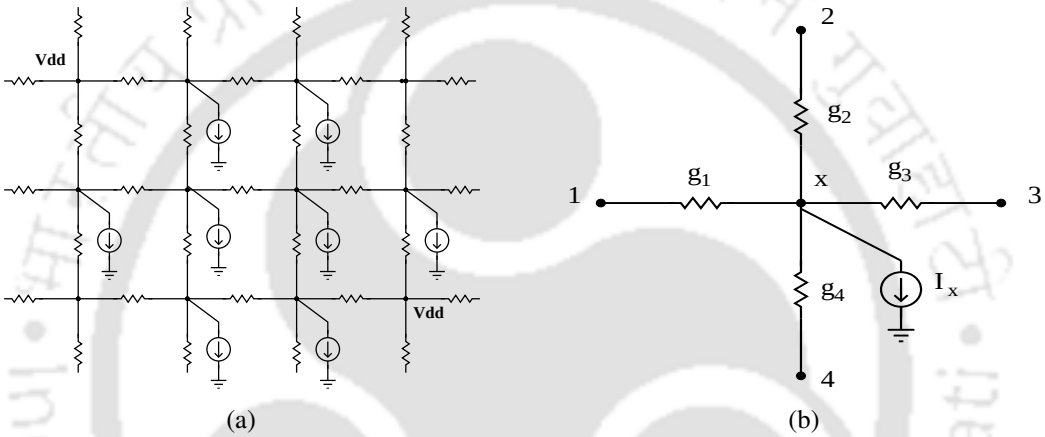


Figure 3.1: (a) A resistive model of power distribution network. (b) A single power distribution node.

work formulates a large linear system with a conductance matrix **A** modeling impedances, an unknown voltage vector **x** modeling the node voltages and a right hand side vector **b** modeling the independent sources, i.e., $\mathbf{Ax} = \mathbf{b}$. Since the model of power distribution network consists of resistances, voltage sources (with one end connected to ground node) and current sources, we use nodal or modified nodal analysis method to analyze power distribution network. Modified nodal analysis method is not suitable for this purpose because it generates a system matrix **A** of size $(n + V - 1) \times (n + V - 1)$, where n and V are the number of nodes and number of voltage sources present in the circuit, respectively, and the system matrix **A** also has a zero diagonal sub-matrix, which makes it unsuitable to be used by iterative solvers. Since all the voltage sources in power grid network are connected to the ground, nodal analysis method can be used to analyze such networks. The system matrix **A** of the

order $(n - V - 1) \times (n - V - 1)$ generated by nodal analysis for power grid network is symmetric and positive definite in nature [106].

Theorem 1. *The system matrix* $\mathbf{A}$ *generated by nodal analysis of the linear system in power grid network formulation, is of the order of* $(n - V - 1) \times (n - V - 1)$*, where* n *and* V *are the total nodes and total voltage sources in the circuit, respectively.*
Notations used : $\mathbf{G}$ *(diagonal matrix),* $\mathbf{i}$ *(current vector)*

Proof. Step 1:

$$\begin{aligned} \mathbf{A}_r\mathbf{i} &= 0 \\ \mathbf{A}_r\mathbf{i} &\equiv [\mathbf{A}_{rG}\ \mathbf{A}_{rJ}]\,\mathbf{i} \\ \begin{bmatrix}\mathbf{A}_{rG} & \mathbf{A}_{rJ}\end{bmatrix}\begin{bmatrix}\mathbf{i}_G \\ \mathbf{i}_J\end{bmatrix} &= 0 \\ \mathbf{A}_{rG}\mathbf{i}_G &= -\mathbf{A}_{rJ}\mathbf{i}_J \ \ (KCE) \end{aligned} \tag{3.2}$$

Step 2:

$$\begin{aligned} Let\ \mathbf{i}_G &= \mathbf{G}\mathbf{v}_G \ \ (Device\ Characteristics) \\ Then,\ \mathbf{A}_{rG}\mathbf{G}\mathbf{v}_G &= -\mathbf{A}_{rJ}\mathbf{i}_J \end{aligned} \tag{3.3}$$

Step 3:

$$\begin{bmatrix}\mathbf{v}_G \\ \dots \\ \mathbf{v}_J\end{bmatrix} = \begin{bmatrix}\mathbf{A}_{rG}^T \\ \dots \\ \mathbf{A}_{rJ}^T\end{bmatrix}\mathbf{v}_n \ \ (KVE) \tag{3.4}$$

Substituting (3.4) in (3.3), we have,

$$(\mathbf{A}_{rG}\mathbf{G}\mathbf{A}_{rG}^T)\mathbf{v}_n = -\mathbf{A}_{rJ}\mathbf{i}_J \tag{3.5}$$

□

It can be proved that the size of system matrix $\mathbf{A} = \mathbf{A}_{rG}\mathbf{G}\mathbf{A}_{rG}^T$, can be $(n-1)\times(n-1)$ [106], where $\mathbf{G}$ is the diagonal matrix and $\mathbf{A}_{rG}$ and $\mathbf{A}_{rJ}$ denote the reduced incidence matrices corresponding to conductance type branches and current source type branches, respectively. For the nodes of power grid network, whose potentials are known, it is not necessary to formulate Kirchhoff current equations (KCE). This will reduce the effective number of power grid nodes to $n - V - 1$, for which KCE equations need to be formulated, where V represent the nodes whose potentials are known. Thus, the size of system matrix being generated from the analysis of power distribution network becomes $(n - V - 1) \times (n - V - 1)$.

3.2.2 Network Formulation for Transient Analysis

For transient analysis of power distribution network, we consider the effects of capacitances and time-varying current waveforms. In this section, during implementation, the backward Euler approximation with timestep h is considered to transform the differential equations to linear equations. We assume timestep h to be constant during transient analysis of power distribution network. The capacitances are modeled to be connected to ground along with the metal resistances of the power distribution network, which is shown in the Figure 3.2(a). The current sink used in the transient formulation is a time-varying current source.

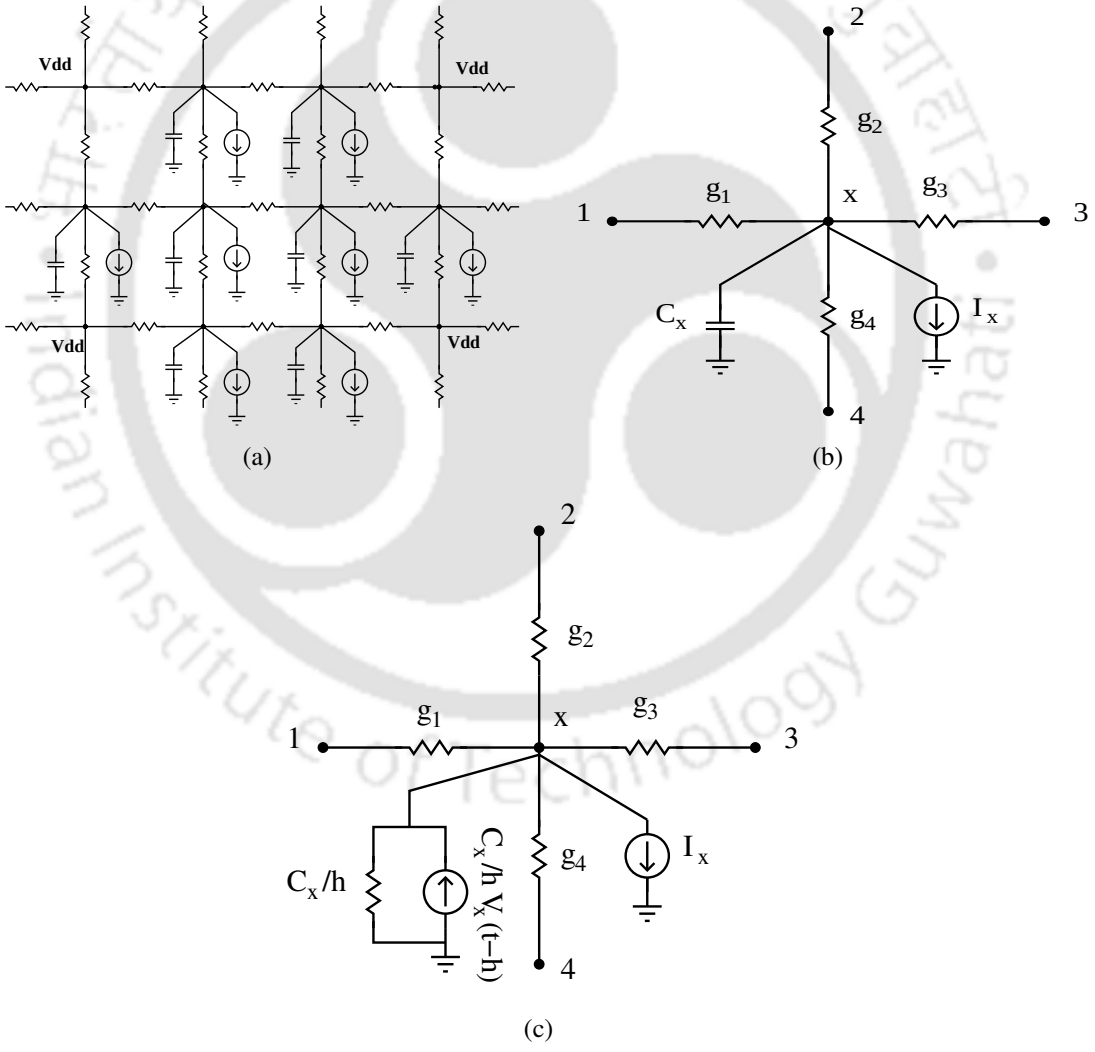


Figure 3.2: (a) RC modeled power distribution network, (b) A single power distribution node, (c) power distribution node with Norton representation of backward Euler model for capacitor C_x.

Therefore, the system of equations for RC-modeled power distribution network can be written as follows,

$$\mathbf{A}\mathbf{x}(t) + \mathbf{C}\,\mathbf{x}'(t) = \mathbf{b}(t), \tag{3.6}$$

where $\mathbf{A}$ is the conductance matrix; $\mathbf{x}(t)$ is the vector of time-varying node potentials; $\mathbf{x}'(t)$ is the vector of rate of change of the node potentials; $\mathbf{C}$ is the impedance matrix corresponding to the capacitances of the power distribution network, and $\mathbf{b}(t)$ is the time-varying current sinks. Using backward Euler's method with a timestep h, the linearized form of system of equations can be written as,

$$(\mathbf{A} + \frac{\mathbf{C}}{h})\mathbf{x}(t) = \mathbf{b}(t) + \frac{\mathbf{C}}{h}\mathbf{x}(t-h) \tag{3.7}$$

Applying Kirchhoff's current law and Kirchhoff's voltage law at a node x (as shown in Figure 3.1(b)), we have [82],

$$\sum_{y=1}^{k} g_y\,(V_y(t) - V_x(t)) = \frac{C_x}{h}\,(V_x(t) - V_x(t-h)) + I_x(t), \tag{3.8}$$

where g_y, V_y, I_x and V_x are as defined in (3.1); C_x represents the capacitance between node x and ground. For a network having RC components, where capacitors are connected between two non-ground nodes, those capacitors can be replaced by a network component having resistors and current sources connected in parallel (Norton equivalent as shown in Figure 3.2(c)) or a network component having resistors connected in series with voltage sources (Thevenin equivalent). In this section, we discuss the case described in (3.8). Equation 3.8 can be rewritten as,

$$V_x(t) = \sum_{y=1}^{k} \frac{g_y}{G_x^{total}} V_y(t) + \sum_{y=1}^{k} \frac{\frac{C_x}{h}}{G_x^{total}} V_x(t-h) - \frac{I_x(t)}{G_x^{total}}, \tag{3.9}$$

where $G_x^{total} = \sum_{y=1}^{k} g_y + \frac{C_x}{h}$, g_y is the conductance of y^{th} connected nodes; k represents the number of adjacent nodes of x; V_y is the voltage of the y^{th} adjacent node; I_x is the current drawn from node x, and V_x represents the voltage at node x.

3.3 Random Walk

3.3.1 Application of Traditional Random Walk Algorithm

If we relate the walk of drunkard in a city with the solution of a node potential in the power distribution network, then (2.5) can be compared with (3.1). In that case, nodes, V_{DD} PADs and potential of each node in the power distribution network correspond to the cross sections of the city, the homes of the drunkard and the average of the reward earned by drunkard, respectively. Further, the probability of a walk traversing from node x to the adjacent node y can be represented as [82],

$$p_{x,y} = \sum_{y=1}^{k} \frac{g_y}{\sum_{y=1}^{k} g_y}, \tag{3.10}$$

and the amount paid at each cross-section can be expressed as,

$$r_x = \frac{I_x}{\sum_{y=1}^{k} g_y}, \tag{3.11}$$

where, (3.10) and (3.11) represent the probability and penalty to be paid during steady state analysis of power distribution network, respectively. However, during transient analysis, the evaluation of probability and reward of the random walk are changed to accommodate (3.9). In this case, it is assumed that each node x has an additional branch connected to a node. This node can be regarded as a virtual home node and the walk may end on this node with a probability,

$$p_{x,y} = \sum_{y=1}^{k} \frac{\frac{C_x}{h}}{G_x^{total}}, \tag{3.12}$$

and reward to the drunkard may be estimated as $V_x(t-h)$ [82]. The probability to choose any other node can also be evaluated using (3.10) provided that the amount to be paid at each cross-section can be expressed as,

$$r_x = \frac{I_x(t)}{G_x^{total}}. \tag{3.13}$$

The drunkard can take any forward or backward path to complete his/her walk. As mentioned above, by taking the average of the number of walks in different directions to reach at home, the reward in random walk method is evaluated using (3.11) and (3.13) during steady state

analysis and transient analysis, respectively. Similarly, node potentials can be evaluated by using the same equations treating nodes as the cross-overs and edges as the paths in the power distribution network. However, to speed up the evaluation and to reduce the average number of steps for each walk, cross-sections (nodes) with known costs (potentials) can be regarded as new destinations (homes) for the drunkard [82].

3.3.2 Two-Step Random Walk (TSRW) Method

One of the major drawbacks of random walk algorithm is that the degree of time complexity is limited by the average walk length of the random walk [82]. With the increase in the walk length and size of power distribution network, the degree of time complexity increases linearly. The objective of Two-Step Random Walk (TSRW) method is to reduce the walk length up to a certain limit by analyzing power distribution network in two steps. To begin the process, entire power distribution network is divided into small horizontal blocks of equal size called subcircuits as shown in Figure 3.3(a). In the first step, boundary nodes of all the subcircuits are solved by random walk method and these boundary nodes are marked as new homes. Later in the second step, the nodes inside the subcircuits are solved by the same random walk method. In this way, walk length is reduced in a significant manner, which in turn speeds up the solution process. In the second step, the random walks are confined within each subcircuit and each walk is terminated as soon as home nodes are reached. This reduces average walk length in each subcircuit. The details of the proposed algorithm is explained in the form of a flowchart in Figure 3.3(b) and the pseudocode is described in Algorithm 4 (refer section 3.3.2.2).

3.3.2.1 Scalability Analysis

Scalability of a parallel system is a measure of its capacity to increase the speed up proportional to the number of processors [107]. To examine the capability of an algorithm for utilizing an increasing number of processing resources effectively on a parallel architecture, several metrics are used, such as isoefficiency analysis [108]. Isoefficiency function of a par-

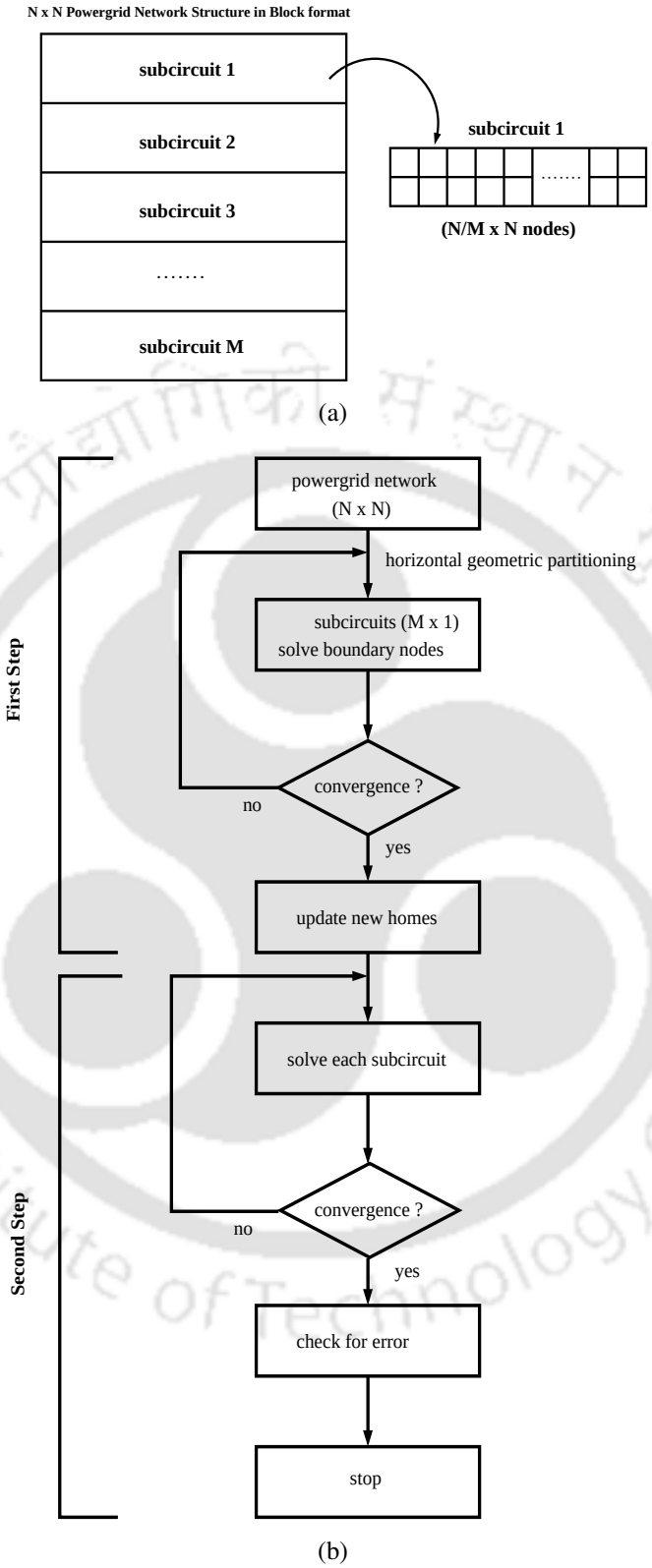


Figure 3.3: (a) Horizontal geometric partitioning, (b) flowchart of TSRW method.

allel algorithm correlates a problem size to a number of processing resources to sustain a fixed efficiency. Efficiency of a parallel system is a performance measure of scalability. If a parallel system having n number of processors is used to solve a problem instance of fixed size H with a total computational overhead of $T_0(H, n)$, then the efficiency E can be represented by (3.14),

$$E = \frac{H}{H + T_0(H, n)} \tag{3.14}$$

We have evaluated TSRW method for different number of threads to examine the scalability of this method. The result of this experiment is shown in Figure 3.4. In TSRW approach, an entire power distribution network of size $N \times N$ is decomposed into rectangular $M \times 1$ subcircuits[1], each having $\frac{N}{M} \times N$ nodes. The proposed TSRW approach requires each task to have at least one subcircuit. Therefore, maximum 32 processing elements (hardware threads) can be used productively when n=32 or 64 processing elements (hardware threads) when n=64 to maintain an efficiency (E) of 50%. As there are 32 logical cores available in Intel Xeon processor, it is observed that performance of TSRW method is optimum in terms of percentage error for 32 threads on this processor while analyzing different power distribution networks[2].

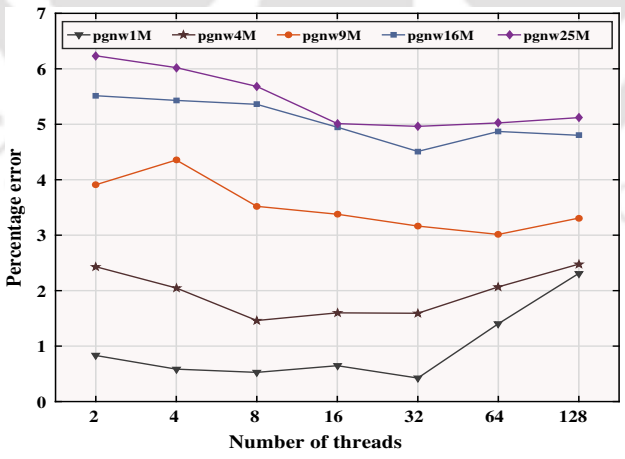


Figure 3.4: Scalability Analysis of proposed TSRW method for different power distribution networks (i.e. pgnw1M (1 million), pgnw4M (4 million), pgnw9M (9 million), pgnw16M (16 million) and pgnw25M (25 million)). Variation of percentage error with the increase in number of threads.

[1] Here each power distribution network is decomposed into 10×1 subcircuits during implementation of TSRW method.

[2] Details on power distribution network benchmarks are provided in Appendix A.

3.3.2.2 Implementation Details

The implementation of parallel TSRW algorithm for steady state analysis of power distribution network is described below along with the pseudocode of TSRW, which is exhibited in Algorithm 4.

(i) *Reading input file:* At first, the circuit is read from the input file provided in SPICE netlist file format. After that all other necessary informations for every node such as neighbors, path conductances, current source values and types of the nodes are extracted SPICE netlist.

(ii) *Initial setup:* The probability is evaluated for every adjacent nodes connected with each node in the power distribution network along with the rewards or pay off values using (3.10) and (3.11). If the node is a V_{DD} PAD, then it is treated as a home and the potential at each node in the path is treated as the average value of the reward.

Algorithm 4: Two-Step Random Walk (TSRW)

Input : A power grid network of size $N \times N$.
Output: Node potential at each node $i \in P^{N \times N}$.
Data: $i \in B^n$, such that $B^n \subseteq S$ and $B^n \cup I^n = S$, where S is a set of all subcircuits of power grid network $P^{N \times N}$. B^n is set of boundary nodes and I^n is set of inner nodes of each subcircuit n respectively.

1: $V_{boundaryNode} \longleftarrow 0$
2: $V_{innerNode} \longleftarrow 0$
3: #pragma omp parallel
4: **for** $i \in B^n$ **do**
5: `RANDOMWALK(`i`)` ▷ `solving boundary nodes`
6: find $V^i_{boundaryNode}$
7: **end for**
8: #pragma omp parallel
9: **for** $k \in I^n$ **do**
10: `RANDOMWALK(`k`)` ▷ `solving inner nodes`
11: find $V^k_{innerNode}$
12: **end for**

(iii) *Starting the walk:* After setting the initial requirements, random walk is started from any node chosen at random. As the walks are independent, it can be started simultaneously from many nodes in parallel. Once the walk gets completed for a node, the node

Algorithm 5: `RANDOMWALK(`x`)`

```
#pragma omp parallel
for w ← 1, no_of_walks do
    total_reward ← 0
    while node_type ≠ home do
        pick a rand_num
        for k ← 1, N_neighbor do
            if rand_num < probability then
                break    ▷ probability is evaluated using (3.10)
            end if
            find node_id of k^th node
            update neighbor
            update total_reward
        end for
        update node_potential
    end while
    node_potential ← node_potential walks
    node_type ← home
    reward ← node_potential
end for
if stopping_criteria met then
    break
end if
```

is assigned to be a new home and the reward value becomes equal to the potential of the node calculated by the random walk. As the number of completed walks grows, the number of home nodes increases in the network and the computational time required to complete the walk for subsequent nodes reduces significantly.

(iv) *Stopping criteria:* If V_x is the walk gain from node x and W is the number of walks from node x, the estimate of variance V_x can be represented as,

$$\sigma_x^2 = \frac{1}{W-1}\sum_{w=1}^{W}(V_x^w - v_x')^2 \tag{3.15}$$

The error is estimated for each solution v_x, where x is the starting node of walk. This error should be smaller than a given threshold $\triangle$.

$$-\triangle < v_x - v_x' < \triangle \tag{3.16}$$

where v_x' is the estimate of solution v_x.

3.4 River Formation Dynamics (RFD)

River formation dynamics algorithm begins with the initialization of water drops to starting nodes (high altitudes). The path followed by each water drop from high altitude to search destination is chosen probabilistically and is administered through evaluating downstream slope at each step. The applicability of RFD to analyze NP-complete problems has been shown in literature [84], [109]. The principle of RFD scheme can be suitably explained by taking an example of a graph Θ having N number of nodes (say $N = \{n_1,\ n_2, \ldots, n_N\}$) and E number of edges. The RFD algorithm as described in [84] starts with initialization of drops to the initial node (say n_1) and then, initialization of all the nodes of graph Θ, i.e. the altitudes of all starting nodes are set to an initial guess and the altitudes of all destination nodes (*sea*) are set to zero. Water drops move from one node to next until the destination (sea) is reached. The transition of a drop from any node to one of the neighbors is occurred probabilistically [84] by following (3.17),

$$P^x_{(n_i,n_j)} = \begin{cases} \frac{\nabla Dr(n_i,n_j)}{\sum_{n_k=1}^{\deg(n_i)} \nabla Dr(n_i,n_k)} & if\ n_j \in N^x_r(n_i), \\ 0 & if\ n_j \notin N^x_r(n_i), \end{cases} \tag{3.17}$$

where $\deg(n_i)$ is the number of connected nodes and $N^x_r(n_i)$ denotes the set of nodes that are neighbors of node n_i. $\nabla Dr(n_i, n_j)$ represents the gradient between node n_i and n_j and can be defined by (3.18),

$$\nabla Dr(n_i, n_j) = \frac{V_{alt}(n_i) - V_{alt}(n_j)}{W(n_i, n_j)}, \tag{3.18}$$

where $V_{alt}(n_i)$ and $V_{alt}(n_j)$ are altitudes of nodes n_i and n_j, respectively, and $W(n_i, n_j)$ denotes weight (cost) of the edge $e_{(n_i,n_j)}$. Following the geographical metaphor, rivers flow from high altitudes to sea-level and during this process of water flow, continuous removal of soil and debris on the land surface leads to erosion. Therefore, when a drop moves from node n_1 to node n_2, the altitude of node n_1 is eroded (reduced) by the current gradient $\nabla Dr(n_1, n_2)$ and this erosion process is transmitted to the destination node. Once the drop reaches its des-

tination, the algorithm is repeated by inserting new drops at the initial nodes which in turn multiplies the erosion of potential paths. Sometimes due to excessive erosion, the altitude of nodes may become close to zero making the gradient imperceptible. This may affect the already formed paths in due process and may lead to inefficient solutions. To avoid such situations, the altitude of nodes are increased by depositing sediments which helps in unblocking alternate paths from initial nodes. In the end, all possible potential paths are analyzed to select the optimum one.

When there are multiple drops flowing from several initial nodes, it is not necessary that RFD would provide a single best possible solution. It can be considered that instead of a single main stream, a group of tributaries tends to follow several individual paths to reach the destination from each initial node. In this case, it is not imperative that each individual path from origin to destination is the best one, i.e. shortest path. In particular, this behavior of forming multiple individual paths to reach destination can be applied suitably to perform steady state power grid analysis. Water drops can follow individual paths from each node to reach any V_{DD} nodes (destinations or PADs) to complete individual flows by eroding traversed nodes. The amount of erosion at a node is calculated by the gradient between the current and the next node. However, it can also be evaluated by considering gradients of all possible choices at the current node instead of taking a single gradient, which results in a straightforward adaptation to perform power distribution network analysis.

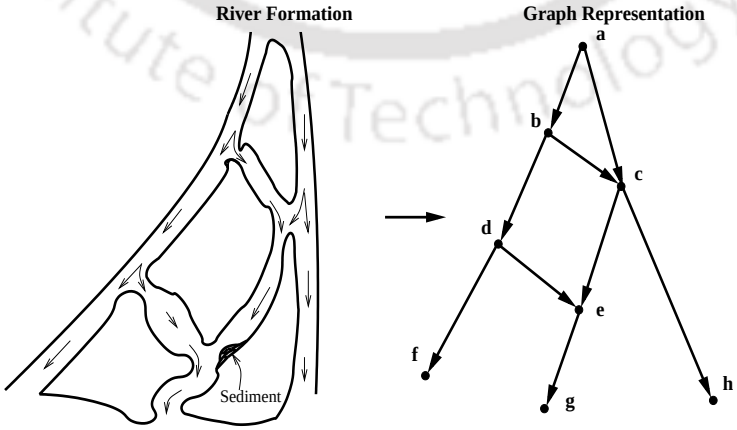


Figure 3.5: Graph representation of RFD scheme.

3.4.1 Application of RFD in the analysis of VLSI power grid networks

For implementation of RFD algorithm, a resistive model of V_{DD} grid is taken as shown in Figure 3.1(a) for steady state analysis and a RC-modeled power distribution network is considered as shown in Figure 3.2(a) for transient analysis. As it is already shown in literature that a power distribution network can be expressed in the form of linear system of equations, the potential of nodes can be obtained by finding best paths traversed by each drop from source to destination. A generic way to approach power distribution network analysis is to allow a number of drops to take different paths and use the average gradient evaluated in each path as an approximated solution. If the gradient value is averaged by taking a sufficient number of water drops, then an acceptably accurate solution can be obtained. A simple graph representation of RFD is shown in Figure 3.5.

3.4.1.1 Graphical Approach

Let Θ be a connected graph having N nodes and E edges. If a drop is allowed to flow in graph Θ, starting at an initial node n_1, then the probability of drop at node n_k after k^{th} transition will be $\frac{1}{d(n_k)}$, where d denotes the dimension of the graph Θ. Let a drop is set to flow on a power distribution network having system matrix, $A_{rG}GA_{rG}^{T}$ of size $(n-1)\times(n-1)$ as described in (3.5). Let T_p be a matrix of transition probabilities such that,

$$T_p = (P_{n_i n_j})_{n_i, n_j \in N} = GA_G, \tag{3.19}$$

where G is the diagonal matrix and A_G is the adjacency matrix corresponding to conductance type branches in power distribution network. The rule of transition can be expressed by [110],

$$P_{n_{k+1}} = T_p^T P_{n_k}, \tag{3.20}$$

$$P_{n_k} = (T_p^T)^k P_0, \tag{3.21}$$

where P_{n_k} denotes probability distribution after k^{th} step. If a drop starts flowing from node n_i, then it reaches node n_j in k^{th} step and the value of $P_{n_i n_j}$ can be calculated by looking into

the $(ij)^{th}$ entry of matrix T^k. Alternatively, harmonic functions [110] can be used to show the adaptability of RFD algorithm to express potential of the nodes across a power grid network. A function $f : V \rightarrow \mathbb{R}$ is said to be a harmonic function with set of poles M if it can be expressed as [110],

$$P_{n_i} \sum_{n_j \in deg(n_i)} f(n_j) = f(n_i) \quad n_i \notin M. \tag{3.22}$$

Let $f(n_i)$ denotes the probability that a drop flowing from node n_i would reach node n_j before node n_k, then, f is said to be a harmonic function having poles n_j and n_k. If Θ can be considered as a graph to power distribution network having unit conductance value along the edges, then $f(n_i)$ denotes the potential at node n_i with current (drop) flowing from node n_j to node n_k.

3.4.1.2 Electrical Circuit Theory Approach

Steady state analysis of power distribution network requires modeling of the grid as an electrical network composed of resistors, source voltages and currents. Figure 3.1(a) represents a single node x of the V_{DD} grid. Application of Kirchhoff's current law at node x results in an algebraic equation (3.23),

$$\sum_{y=1}^{k} I_y = I_x, \tag{3.23}$$

where k is the number of neighbors of x; I_y denotes the current flowing through y^{th} connected node and I_x is the current drawn from node x. Further, each I_y current can be represented by,

$$\sum_{y=1}^{k} I_y = \sum_{y=1}^{k} g_y(V_y - V_x), \tag{3.24}$$

where g_y and V_y denote the conductance and potential of y^{th} adjacent node, respectively, and V_x represents the potential at node x. If we relate the flow of a drop from origin to destination with solution of a node potential in a power distribution network, V_{dd} PADs would correspond to the destinations (sea). Similarly, potential of each node would correspond to altitude of a node after erosion process and weight (cost) of an edge can be correlated to resistance. If $p_{x,y}$ represents probability of a drop flowing in any direction y starting from x, then (3.17) can be

formulated as,

$$p_{x,y} = \frac{\nabla Dr(x,y)}{\sum_{y=1}^{k} \nabla Dr(x,y)}, \tag{3.25}$$

and the gradient between node x and node y can be represented as,

$$\nabla Dr(x,y) = g_y(V_x - V_y) \tag{3.26}$$

Comparing (3.26) and (3.24) and taking their absolute values, the probability that a drop traverses from node x to node y can be represented as,

$$p_{x,y} = \frac{I_y}{I_x} = \frac{g_y(V_y - V_x)}{\sum_{y=1}^{k} g_y(V_y - V_x)} \tag{3.27}$$

Similarly, during transient analysis of power distribution network, the probability $p_{x,y}$ can be represented as,

$$p_{x,y} = \frac{I_y(t)}{I_x(t)} = \frac{g_y(V_y(t) - V_x(t))}{\sum_{y=1}^{k} g_y(V_y(t) - V_x(t)) - \frac{C_x}{h}(V_x(t) - V_x(t-h))} \tag{3.28}$$

3.4.2 Implementation Details

In this subsection, implementation framework of RFD scheme to compute voltage fluctuation for each node is shown in detail to analyze the power distribution network. RFD takes natural advantage of grid locality to improve efficiency in speeding up the analysis to provide way for location estimation of hotspots. One of the reasons for this improved efficiency in terms of memory may be the avoidance of any explicit representation of system matrices corresponding to power distribution networks. In RFD, water drops search for new paths according to altitude gradients of power grid network edges. The search for new edge from each node of power grid network comes with the computation of probability from altitude gradients as described in (3.26). After following an edge, the altitude or potential of the corresponding node is updated by taking into account the amount of erosion. For efficient applicability of RFD to power grid analysis, the amount of erosion at a particular node is considered as the potential across current sink branch of that node (except V_{dd} nodes), which is computed by

using (3.29), i.e.,

$$V_x = -\frac{I_x}{G_x^{total}}, \tag{3.29}$$

where V_x denotes the potential across current sink branch of node x, I_x represents the amount of current flowing through current sink branch of node x and G_x^{total} denotes sum of conductances of each edge connected to node x. The same procedure is repeated until all the drops reach V_{dd} nodes. Once a drop reaches any V_{DD} node across the power distribution network, the altitude or potential of start node is computed by summing up the amount of erosion at visited nodes with the potential of destination. The process of estimating the potential across different nodes of a power grid network is performed until convergence is achieved, i.e., stopping criteria. Here stopping criteria is represented as,

$$-\beta < V_x - V_x' < \beta, \tag{3.30}$$

where V_x' denotes estimate of solution V_x of an initial node x and β represents error threshold. During both steady state and transient analyses, similar procedure is repeated with an exception that the drop can end its traversal in any of the virtual node with probability as described in (3.12) and the amount of erosion corresponding to the drop is estimated as $V_x(t-h)$ during transient analysis. The procedure to perform power distribution network analysis using RFD is described in Algorithm 6.

Although RFD is performed to estimate the location of hotspots by considering all relevant parameters during implementation, the accuracy is affected by few error components. However, error in solution is sustained so that the potentials at each node can be approximated to the closest of millivolts with small inaccuracy. The error is evaluated for each solution V_x by estimating the variance σ_x^2 for N_x number of drops having path gain of D_x from node x.

$$\sigma_x^2 = \frac{1}{N_x - 1} \sum_{n=1}^{N_x} (D_x^n - V_x')^2 \tag{3.31}$$

The error is not optimal if search of a drop follows a path that leads to a loop, i.e., local traps. Often these traps happen in case of complex power grid networks having irregular structures.

Algorithm 6: Procedure of RFD for power distribution network analysis

Input : A power grid network of size $N \times N$.
Output : Node potential (altitude) at each node $n_i \in P^{N \times N}$.

for $i = 1$ **to** $totalNodes$ **do**
  for $j \leftarrow 1, dropVolume$ **do**
    $total\_erosion \leftarrow 0$
    while $node\_type \neq V_{dd}(sea)$ **do**
      $rand\_calculation$
      for $k = 1$ **to** $nodes\_connected$ **do**
        if $rand\_num < Probability$ **then**
          $break$
        end if
      end for
      $find\ drop\_id\ of\ k^{th}\ node$
      $update\ altitude\_connectedNode$
      $update\ total\_erosion$
    end while
    $update\ node\_altitude$
  end for
  $node\_altitude \leftarrow node\_altitude\_withDrops$
  $node\_type \leftarrow V_{dd}$
  $erodeValue \leftarrow node\_altitude$
end for

In such cases, several trap avoiding strategies [82] can be employed for performance enhancement of the proposed methodology.

3.5 Simulation Results

All algorithms including the proposed RFD metaheuristic and TSRW algorithm are implemented by using C/C++ language on Linux environment on a machine with Intel Xeon E5-1603 processor. Both the performance of RFD and TSRW[3] are evaluated on Nvidia Tesla K20c GPU having 2500 CUDA cores approximately and 5GB of global device memory. Serial RFD code is optimized on GPU using NVCC (Nvidia CUDA compiler) from CUDA toolkit 6.5 targeted to improve performance. All necessary environment variables are adjusted

[3] TSRW is proposed as a parallel implementation of RW and it is implemented on GPU for steady state power distribution network analysis only.

focusing on the efficient execution of serial and parallel versions of proposed methods. A well known heuristic like random walk [82] and its variant (hierarchical random walk (hRW)) [21], an efficient direct solver KLU [19] are evaluated for the comparison of results with our proposed metaheuristics. Again both random walk and an efficient iterative solver CG [20] are implemented on GPU platform for comparison of solutions with GPU versions of RFD and TSRW. Various experiments are carried out on different power distribution networks to showcase the effectiveness and performance of both TSRW and RFD. The benchmarks[4] used for experimentation are produced using our in-house power grid generator without loss of any functionality. These benchmarks are already being used in literature [111] to validate performance of many power grid analyzers. The benchmarks for steady state analysis are generated in the SPICE format with a grid size ranging from 1 million to 25 million nodes. The metal resistances on these benchmarks are set to 1Ω like industrial designs. The distribution of V_{DD} PADs are placed randomly across the benchmarks to have a close realization of industrial power grid designs. The potential across these PADs are set to 1.8V and the current sinks are placed on each nodes reasonably with values set to 0.01A except at the PADs.

Table 3.1: Comparison of computational time to perform steady state analysis of power distribution networks.

Nodes	$t_{RW}\ (s)$	$t_{GS}\ (s)$	$t_{hRW}\ (s)$	$t_{KLU}\ (s)$	$t_{RFD}\ (s)$	Speedup (t_{RW}/t_{RFD})	Speedup (t_{GS}/t_{RFD})	Speedup (t_{hRW}/t_{RFD})	Speedup (t_{KLU}/t_{RFD})
pgnw1M	18.84	14.527	9.01	16.607	7.77	2.42×	1.86×	1.15×	2.137×
pgnw4M	102.03	95.453	40.02	156.13	35.05	2.91×	2.72×	1.14×	4.453×
pgnw9M	378.03	269.95	97.23	518.58	69.02	5.47×	3.91×	1.40×	7.515×
pgnw16M	784.20	595.66	206.67	1372.8	140.13	5.59×	4.25×	1.47×	9.8×
pgnw25M	1338.07	1090.99	325.45	3065.21	211.12	6.33×	5.16×	1.54×	14.5×

The proposed RFD metaheuristic is evaluated on several power distribution benchmarks to compare the solutions of steady state analysis with solutions obtained from other metaheuristics, such as random walk (RW) and hierarchical random walk (hRW). It can be observed from Table 3.1 that the speedup of $6.33\times$ and $1.54\times$ have been achieved over the solutions obtained from RW and hRW, respectively, for a power distribution network having 25 million nodes. It

[4]The details on these benchmarks are provided in Appendix A.

is to be noted that RW solver is one of the well known standard power grid analyzers reported in literature [82]. The performance of RW is slow under suitable general conditions on large power grid networks. Due to the back and forth movement of walk [82], the chances of falling into local traps is more as compared to RFD method. Therefore, the order of magnitude of a walk to complete first n steps can be $(\log n)^3$ which is large as compared to RFD [82, 112]. To showcase effectiveness and efficiency of our proposed RFD metaheuristic in terms of run-time, we have compared the solutions obtained using RFD with solutions obtained from a standard iterative solver Gauss-Seidel (GS) and a direct solver KLU. As listed in Table 3.1, the speedup of $5.16\times$ and $14.5\times$ have been achieved over GS and KLU, respectively, while performing experiments on a power grid benchmark having 25 million nodes. It is observed that GS is slow in evaluating large power grid networks because of slow rate of convergence. Similarly, for large size system matrices resulting from large power distribution network, the performance of KLU is more compute intensive as compared to RFD.

As large size of power distribution network places great demand on memory bandwidth and powerful computations, performing scientific calculations to estimate hotspots necessitate the use of high performance computing units, such as GPU. To exhibit efficient propagation of drops to perform realistic simulations on large power grid networks, TSRW and RFD are implemented on Nvidia GPU platform for accelerated convergence. Each GPU has a number of streaming multiprocessors which include many cores that can perform compute intensive tasks by executing threads in parallel. For large power grid networks, RFD is performed to take advantage of GPU programming model (CUDA) that allows to carry out memory transfers between CPU and GPU, and extensive computations (kernels) simultaneously. These kernels are launched to generate a number of threads organized in an array of thread blocks. The number of threads in each block (represented by $blockDim$ variable) is specified at the beginning before launching the kernel. For starting the execution of RFD on GPU, each drop is taken to be a thread in a block associated with unique global index. Each thread combines its thread index $threadIdx$ and block index $blockIdx$ values to represent a data index i, which

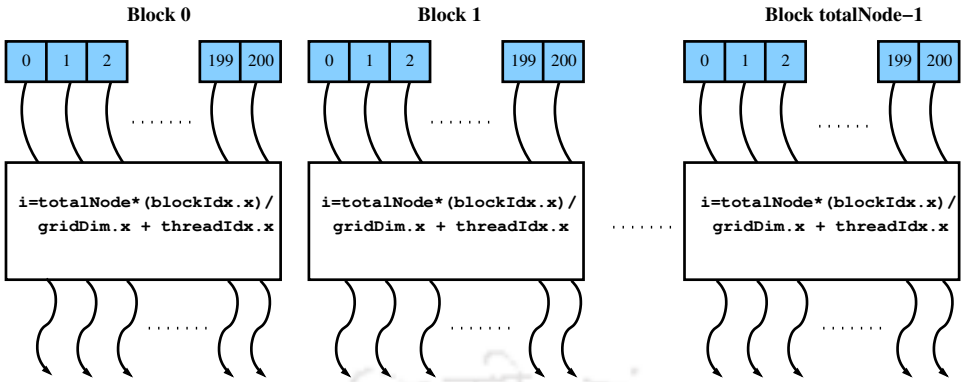


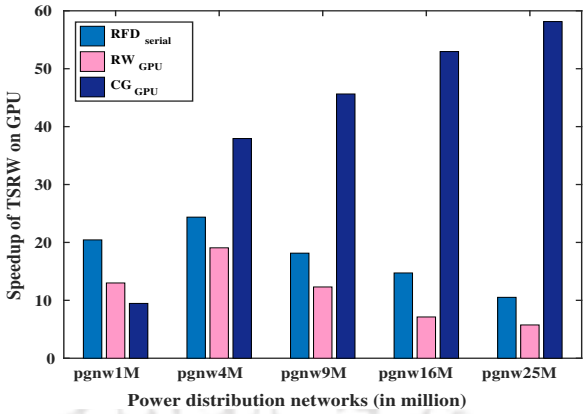
Figure 3.6: Block representation of a kernel.

is calculated as [20],

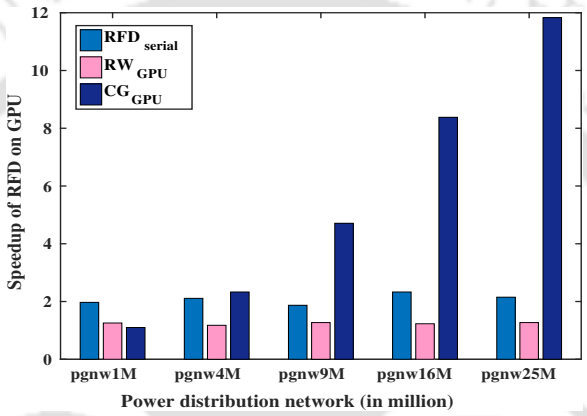
$$i = (blockIdx.x * blockDim.x) + threadIdx.x \tag{3.32}$$

In our implementation, we consider the parameter $(blockIdx.x * blockDim.x)$ as $(blockIdx.x * totalNode)/gridDim.x$ and the grid size is set to 200 for optimum performance as shown in Figure 3.6. During kernel execution of RFD and TSRW , both drops and walks are allowed to propagate across the power distribution network by executing threads of same block for faster convergence, respectively. However, no synchronization methods are employed among threads of each block during execution or after returning the control from GPU to CPU. Only simple kernels are implemented to improve the performances of TSRW and RFD on GPU, i.e., the speedup of $67\times$ and $2.15\times$ have been obtained over serial RW and RFD, respectively, while performing on a power distribution network having 25 million nodes.

It is observed from Figure 3.7(a) and Figure 3.7(b) that the speedup of $5.75\times$ and $1.27\times$ have been achieved over GPU implementation of RW using TSRW and RFD for a power distribution network having 25 million nodes, respectively. It is observed that TSRW has demonstrated much better performance on GPU as compared to RFD. One of the reasons for such performance of RFD on GPU is due to the use of simple kernels and, because of the absence of synchronization methods among threads of same blocks and the inefficient use of helper kernel to gather results after execution of threads on kernel performing calculations. In view of this, a significant milestone has not been achieved in terms of runtime while imple-



(a)



(b)

Figure 3.7: Speedup achieved by (a) TSRW and (b) RFD on GPU over other techniques while performing steady state analysis on different power distribution networks.

mentation of RFD on GPU. However, the performance of RFD on GPU can be improved by employing efficient helper kernels, which can reduce the reverse data transfer penalty from GPU to CPU. The performance of RFD on GPU can also be improved by making use of idle threads while the execution of current thread becomes stalled due to the delay in accessing memory. Further, similar to TSRW, the two-step strategy can also be employed using RFD metaheuristic to improve the performance on GPU. We have also compared the results of TSRW and RFD heuristics with the solutions obtained from CG implemented on GPU platform to showcase the runtime improvement. It can be observed from Figure 3.7(a) and Figure 3.7(b) that both TSRW and RFD on GPU have demonstrated speedup of $58.14\times$ and $11.83\times$, respectively, over CG (on GPU) while analyzing a power distribution network having 25 mil-

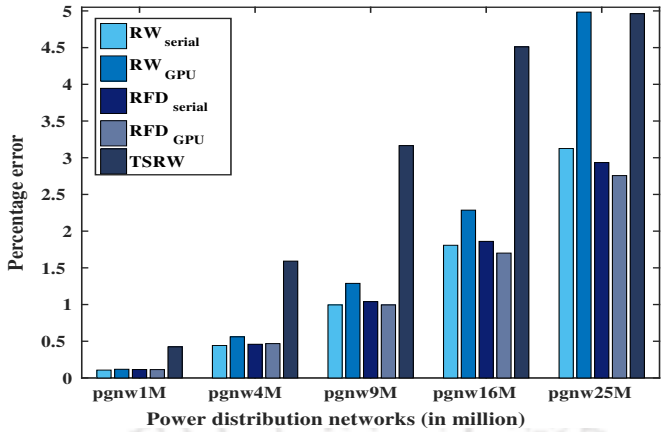


Figure 3.8: Percentage error evaluated during steady state analysis for both serial RW, RW on GPU, serial RFD, TSRW on GPU and RFD on GPU.

lion nodes. One of the reasons for the enormous speedup of TSRW over other algorithms is due to the shorter walk lengths. In serial random walk method, every time an outer loop iterates, nodes having known potentials act as new home nodes, while in TSRW method, multiple random walks take place simultaneously, thus resulting in more number of home nodes at each step. More home nodes at each iteration result in shorter walk lengths for unprocessed nodes, which in turn speeds up the computing performance in a non-deterministic way. A general trend of shorter walk length is also observed after each iteration during the execution of both RW and hRW when nodes having known potentials act as new home nodes, but this process takes place at a slower pace.

Moreover, the residual error (average error) is kept under a satisfactory level of 3% as shown in Figure 3.8 during implementation of RFD metaheuristic, which is found to be considerably less as compared to the performance of RW (both serial and GPU implementations) and the proposed TSRW metaheuristic. With increase in size of the power distribution networks, it becomes a challenging task to keep the error under this satisfactory level. However, RFD demonstrates superior performance in terms of accuracy of solutions and outperforms RW based metaheuristics (RW, $\mathrm{RW}_{\mathrm{GPU}}$ and TSRW) due to an unidirectional movement of drops. This favors the movement of drops to escape from any local traps and reach destination within reasonable computational time.

Table 3.2: Comparison of computational time (on CPU) for transient analysis of power distribution networks.

Benchmark (pgtnw#)[1]	t_{RW} (s)	t_{GS} (s)	t_{hRW} (s)	t_{RFD} (s)	Speedup (t_{RW}/t_{RFD})	Speedup (t_{GS}/t_{RFD})	Speedup (t_{hRW}/t_{RFD})
pgtnw10K	3.52	6.17	3.12	2.67	1.31×	2.31×	1.16×
pgtnw20K	9.49	15.45	6.03	5.10	1.86×	3.02×	1.18×
pgtnw30K	16.91	23.17	10.67	8.85	1.91×	2.61×	1.21×
pgtnw40K	27.68	31.86	19.89	13.21	2.09×	2.41×	1.50×
pgtnw50K	36.94	39.02	24.23	19.56	1.88×	1.99×	1.23×
pgtnw100K	102.11	98.14	59.30	46.85	2.18×	2.10×	1.26×
pgtnw500K	923.78	881.67	352.03	287.38	3.21×	3.06×	1.22×
pgtnw1M	2103.30	-	859.20	730.80	2.87×	-	1.18×
pgtnw4M	8865.02	-	4214.60	2895.00	3.05×	-	1.45×

[1] # denotes the number of nodes in the power distribution network.

Further, transient analysis has been performed on several RC-modeled power distribution benchmarks to demonstrate the effectiveness of RFD metaheuristic. The benchmarks for transient analysis are generated in the SPICE format with a grid size ranging from ten thousand to 4 million nodes. The metal resistances on these benchmarks are set to 1Ω and capacitances are set to 0.1μF similar to industrial designs. V_{DD} PADs having potential set to 1.8V are placed randomly across the benchmarks to have a close realization of industrial power grid designs. The current sinks are placed on each nodes reasonably with values set to 0.01A except at the PADs. The CPU times are measured after completion of all timesteps during implementation. The results of different algorithms are compared and listed in Table 3.2. It is observed from Table 3.2 that the speedup of $3.21\times$, $3.06\times$ and $1.22\times$ have been obtained over RW, GS and hRW algorithms, respectively, while analyzing a power distribution network having $500,000$ nodes (number of timesteps considered within a time period of 1ms is 500). With further increase in size of power distribution networks, it is observed that the iterative solver GS fails to provide solutions for two benchmarks (pgtnw1M and pgtnw4M) and RFD demonstrates speedup of $3.05\times$ and $1.45\times$ over RW and hRW algorithms, respectively, while analyzing pgtnw4M benchmark. The average residual error is kept under a satisfactory level

of 3% (with a maximum error of $10m$V being observed while solving pgtnw4M benchmark) during experiments.

3.6 Summary

In this chapter, we have presented a fast, simple and an efficient parallel power grid analysis method based on random walk algorithm to solve large power grid networks on GPU. The proposed approach requires only few steps to find a solution and it can be used to compute conservative bounds on the worst-case voltage drops in power distribution network. The speedup of 10.52$\times$ and 5.75$\times$ have been obtained using TSRW method while solving a power distribution network having 25 million nodes over serial RFD and GPU version of RW, respectively. Apart from TSRW method, an efficient parallel power grid analysis method based on RFD is also presented to solve large power distribution networks. Applicability of the proposed RFD method to perform both steady state and transient power distribution network analyses is shown by analytical formulation using graphical and electrical circuit theory approaches. The idea of both serial and GPU implementation of RFD is conveyed to showcase the effectiveness in runtime and accuracy while performing steady state analysis. Although significant runtime of RFD has not been achieved on GPU, the residual error is kept under a satisfactory level of 3%. As it is quiet a challenging task to perform transient analysis due to the involvement of parasitic capacitances, only serial version of RFD is implemented and the solutions are compared with RW, GS and hRW algorithms. It is observed that RFD is efficient both in terms of runtime and accuracy than standard heuristics like random walk and its variants, and GS for transient analysis. One of the main reasons for this improved performance of RFD over other methods is due to the unidirectional movement of drops across power distribution network. This favors the movement of drops to avoid local traps and reach the destination (V_{DD}) within reasonable time. As the motivation of the thesis is to demonstrate the applicability of metaheuristics, the application of RFD is extended to perform optimization of power distribution networks in the following chapter.

4

Power Distribution Network Design Optimization

Contents

4.1 Introduction

The design of power distribution networks is crucial to the correct functioning of modern high speed integrated circuits. With rapid increase in the clock frequency and smaller feature sizes, the integrated circuits have become more susceptible to supply voltage fluctuations. With lower supply voltages, smaller voltage drops occur across power rails and these drops become more significant with increase in the size of power distribution network. This affects functionality of the integrated circuit leading to logic failure. Apart from voltage drop across the power rails, there are several effects that influence the circuit functionality, such as noise due to inductive effects, electromigration effects, etc. Usually, designers attempt to circumvent such adverse effects by over-constraining the design of power distribution network through additional routing and placing supplementary boundaries. This increases the overall area of power distribution network and the design space of the entire system becomes congested for other routing and placement activities. Therefore, minimizing the wire area (metal area of power rails) is one of the primary concerns while designing a robust network for power distribution inside an integrated circuit.

In this chapter, we present a nature inspired metaheuristic, i.e., RFD to minimize both IR drop and wire area of large power distribution networks. RFD is employed to demonstrate the applicability in optimization of industry standard power distribution network benchmarks (i.e., IBM benchmarks [113] and industry benchmarks[1]). Further, we introduce additional nature-inspired factors to RFD scheme and formulate a generalized model of the heuristic. These additional factors (i.e., transverse slope statistics and sediment transport rate) are incorporated to control movement of the water drops for better convergence and stability of RFD. Various experiments are performed on single objective test functions and industry standard benchmarks to demonstrate the effectiveness of modified RFD (MRFD) method. It is observed that MRFD outperforms peer metaheuristics while solving various standard single objective test functions and also exhibits better performance in terms of convergence than

[1]Benchmarks provided by Semiconductor Lab, Chandigarh, INDIA

integer programming method, like SLP in optimizing large power distribution networks.

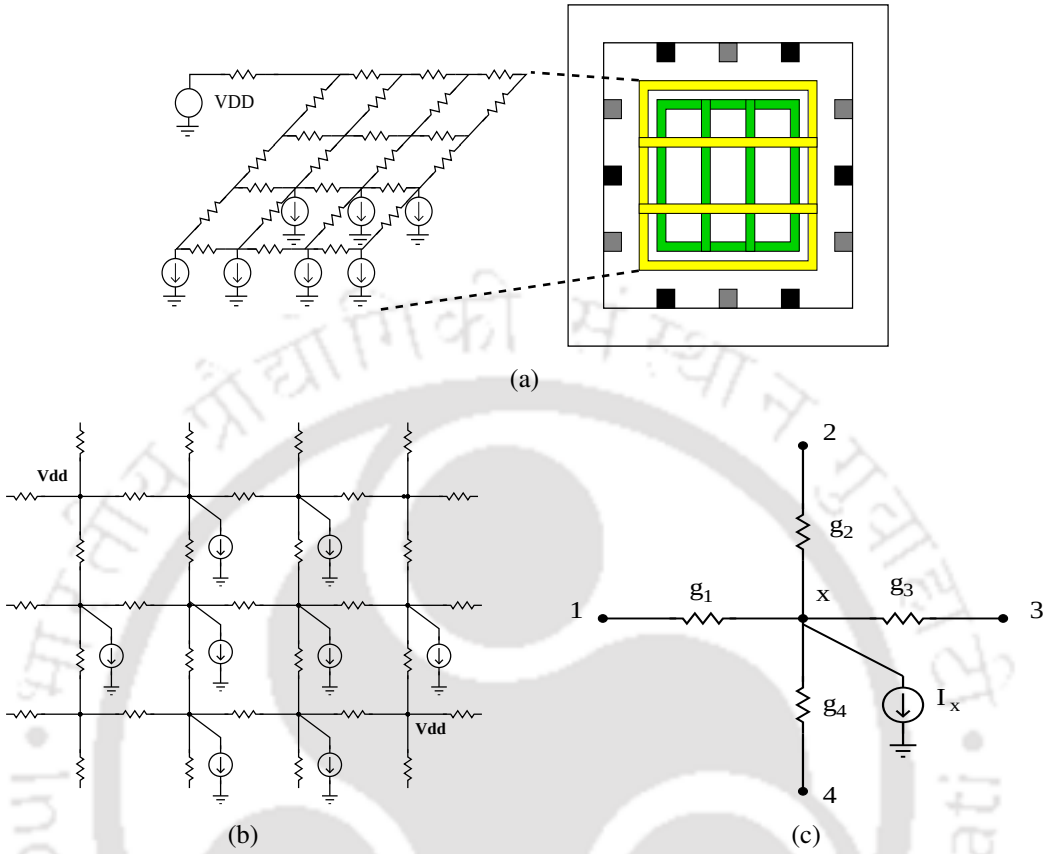


Figure 4.1: (a) Modeling power lines to resistive network. (b) A resistive model of power distribution network. (c) A single power distribution node [32].

4.2 Problem Formulation

4.2.1 Problem 1 : Minimum IR drop

The magnitude of voltage drops and variations depend on the location and placement of the pads as well as wire width. Consequently, the amount of IR drop at a particular position substantially affects the nearby region. Therefore, the IR drop minimization problem is formulated as a mixed-integer nonlinear programming problem. Here, objective is to minimize IR drop of the entire power distribution network by considering each hotspot region individually subject to area, effective width and current constraints. To formulate the IR drop problem, power distribution network needs to be modeled as a network having a set

of horizontal and vertical wires with DC loads connected to V_{DD} power supply as shown in Figure 4.1. To demonstrate the amount of current drawn by the transistor circuits, current sinks are placed at each cross sections except at the power supply nodes. For steady state analysis (DC analysis), entire power distribution network can be formulated as a system, $\mathcal{G}\mathbf{V} = \mathbf{S}, \quad (\mathbf{I}, \mathbf{V}) \in \mathbb{R}_{m\times 1},\ \mathcal{G} \in \mathbb{R}_{m\times m}$, with a conductance matrix $\mathcal{G}$ modeling conductances, an unknown voltage vector $\mathbf{V}$ modeling potential of nodes and a right hand side vector $\mathbf{S}$ modeling independent sources (current sinks and power supply nodes). We can employ Nodal Analysis to analyze the system as described in section 3.2.1. For the power distribution network, system of equations governed by Nodal Analysis as described in (3.5) can be rewritten as (4.1) [106],

$$\left(\mathbf{A}_{r\mathcal{G}}\mathbf{G}\mathbf{A}_{r\mathcal{G}}^{T}\right)\mathbf{v}_n = -\mathbf{A}_{rJ}\mathbf{i}_J + \left(\mathbf{A}_{r\mathcal{G}}\mathbf{G}\mathbf{A}_{r\mathcal{G}}^{T}\right)\mathbf{V}_{\mathrm{dd}}, \tag{4.1}$$

where, $\mathbf{A}_{r\mathcal{G}}$ and $\mathbf{A}_{rJ}$ denote reduced incidence matrices corresponding to conductance and independent current sinks, respectively; $\mathbf{V}_{\mathrm{DD}}$ denotes the vector of power supply nodes (known potential), and $\mathbf{v}_n$ and $\mathbf{i}_J$ denote vectors of node potential and current sinks, respectively. If $\mathbf{V}_{ir}$ is the vector of voltage drop (IR drop) at each node, then (4.1) can be rewritten as (4.2),

$$\begin{aligned}\mathbf{V}_{ir} =& \mathbf{V}_{\mathrm{DD}} - \mathbf{v}_n \\ =& \mathbf{V}_{\mathrm{DD}} - \left(\mathbf{A}_{r\mathcal{G}}\mathbf{G}\mathbf{A}_{r\mathcal{G}}^{T}\right)^{-1}\left(-\mathbf{A}_{rJ}\mathbf{i}_J + \left(\mathbf{A}_{r\mathcal{G}}\mathbf{G}\mathbf{A}_{r\mathcal{G}}^{T}\right)\mathbf{V}_{\mathrm{DD}}\right)\end{aligned} \tag{4.2}$$

While formulating the minimization problem, topology and physical locations of power supply PADs (pins and bond wires) are assumed to be fixed. To capture the impact of IR drop on the performance of power distribution network, a posynomial function f defined in space $\mathcal{C}$ can be represented as (4.3),

$$f(\mathbf{V}_{ir}) = \min_{\mathbf{V}_{ir}\ \in\ \mathcal{C}} \mathbf{V}_{ir} = \min_{\{\gamma_x,\delta_x,v_x\}\in\mathcal{C},\ t\in\mathbb{N}} \sum_{x=1}^{N}(\gamma_x + t\delta_x)\vartheta_x, \tag{4.3}$$

where, N is the number of nodes in power distribution network and γ_x is a nonnegative integer which spans $\mathbf{V}_{ir}$ in the finite-dimensional subspace, such that each IR drop element $\vartheta_x \in \mathbf{V}_{ir}$

is linearly independent. However, if IR drop elements are linearly dependent, then there exists scalars $\delta_1, \ldots, \delta_x$, $x \in N$ such that it satisfies (4.3) and an arbitrary scalar $t \in \mathbb{N}$ can be used to express any IR drop element $\vartheta_x \in \mathcal{C}$. Further, with increase in $t = 0$ to any arbitrary value, there exists an initial value at which $\gamma_h + t\delta_h = 0$ for any index h. This suggests that space $\mathcal{C}$ can be decomposed as (4.4),

$$\mathcal{C} = \bigcup_{h=1}^{N} \left[\sum_{x \neq h} \eta_x \vartheta_x : \eta_x \geq 0, x \neq h \right]. \tag{4.4}$$

Equation (4.4) supports a stronger claim that every IR drop element $\vartheta_x \in \mathcal{C}$ can be represented as a positive combination of a linearly independent subset of $\mathbf{V}_{ir}$ since $\mathcal{C}$ itself is shown to be closed by a finite union of closed sets. In this regard, the optimal IR drop at different nodes of power distribution network can be estimated through the minimization problem defined in (4.3) subject to constraints as shown in (4.5).

$$\text{subject to} \begin{cases} M_{area} \leq \sum_{(x,y)\in R} \rho l_{xy}^2 g_{xy}, \\ \rho l_{xy} g_{xy} \geq w_{min}, \\ \sum_{(x,y)} \frac{i_{xy}}{w_{xy}} \leq \mathcal{J}_{max}, \\ \sum_{(x,y)} (v_y - v_x) g_{xy} \geq i_x, \\ V_{dd} - v_x \leq V_{th}, \ \ \forall x \in N, \end{cases} \tag{4.5}$$

where, M_{area} denotes the area of power distribution network; ρ denotes the sheet resistance; l_{xy}, w_{xy}, g_{xy} and i_{xy} denote length, width, conductance of and current in any branch (x, y), respectively; w_{min} denotes the minimum metal width of any branch; $\mathcal{J}_{max}$ denotes the maximum current density, and V_{th} denotes the threshold of IR drop. As both metal widths of wires (w_{min}) and wire area (M_{area}) have statistical measures on the changes in IR drop profile across the power distribution network [32], they are considered as constraints to bound the amount of IR drop. Further, the effect of current density ($\mathcal{J}$) on the IR drop node under consideration can not be fully ignored and it can be considered as a limit of absolute potential difference of two branches [31]. During analysis each node should follow Kirchhoff's cur-

rent law and it can be expressed as a nonlinear inequality constraint. Further, the IR drop at any node is bounded by a voltage drop constraint, threshold of IR drop (V_{th}). Formulation of the IR drop objective function in (4.4) requires knowledge of voltage profile of the power distribution network.

However, with increase in the size of power distribution network, the size of coefficient matrix $\mathbf{A}_{r\mathcal{G}}$ in (4.2) grows quadratically ($O(n^2)$) (along with IR drop also as shown in Figure 1.4(b)) and the matrix becomes weakly diagonal dominant and irreducible [114]. Deterministic approaches discussed in aforementioned work are no longer reliable in terms of computational time and accuracy to perform power distribution network analysis for minimizing IR drop. In view of this, a heuristic approach, like MRFD can be employed to analyze power distribution network for minimizing IR drop of entire power distribution network efficiently without formulating any system matrix.

4.2.2 Problem 2 : Minimum Metal Area

The basic philosophy of minimizing the wire area of an over-designed PDN using RFD and MRFD methods is to try altering the wire widths without violating maximum voltage drop (IR drop) constraint. As the magnitude of voltage drops and variations depend on wire width, the problem of minimization of wire area can be formulated as follows [32],

$$\begin{aligned} \texttt{Minimize Area} &= \sum_{(x,y)\in E_{xy}} l_{xy} w_{xy}, \\ &= \sum_{(x,y)\in E_{xy}} \rho l_{xy}^2 g_{xy}, g_{xy} = w_{xy}/\rho l_{xy} \end{aligned} \tag{4.6}$$

$$\text{subject to} \begin{cases} \rho l_{xy} g_{xy} \geq w_{min}, \\ \sum_{(x,y)} \frac{i_{xy}}{w_{xy}} \leq \mathcal{J}_{max}, \\ \sum_{(x,y)} (v_y - v_x) g_{xy} \geq i_x, \\ V_{dd} - v_x \leq V_{th}, \ \ \forall x \in N, \end{cases} \tag{4.7}$$

where ρ denotes the sheet resistance; l_{xy}, w_{xy}, g_{xy} and i_{xy} denote length, width, conductance of and current in any branch (x, y), respectively; E_{xy} denotes the set of indexes of all branches in PDN; w_{min} denotes the minimum metal width of any branch; $\mathcal{J}_{max}$ denotes the maximum current density; v_x denotes the voltage at node x; V_{th} denotes the threshold voltage and N is the total number of nodes in PDN.

As the amount of IR drop is crucial to the design of PDN, it is considered as an inequality constraint and is bounded within certain threshold (V_{th}) while wire widths are interchanged to have a minimum wire area during the process of optimization. Further, with changes in width and current value in any branch, current density ($\mathcal{J}$) also changes and can be considered as a limit of absolute potential difference of two branches [31]. During optimization, with change in current value and width of any branch, the total current entering or leaving a node may change accordingly, which imposes a nonlinear inequality constraint on the sum of currents entering or leaving a node (or junction) according to Kirchhoff's current law.

4.3 Minimization using River Formation Dynamics Scheme

4.3.1 River Formation Dynamics

River Formation Dynamics (RFD) is a heuristic based on random probabilistic movement of water drops [84]. Water drops travel downhill to reach destination (sea) by following a number of paths and these paths are chosen probabilistically using a decreasing gradient principle. As a water drop follows a random path of decreasing altitudes, gradients are evaluated in each step by calculating downward longitudinal slopes among different altitudes. The path having the steepest downslope is observed to have better probability and is chosen as the path to be followed in the next step by water drop. The process is continued until water drop reaches destination. In view of this, the number of possible paths from point of origin to destination is evaluated for each water drop. Finally, all the followed paths are analyzed and one potential path is chosen as the best possible solution.

The principle of RFD scheme is described in Algorithm 7. As the search process of

RFD is drawn stochastically to follow a decreasing gradient direction, the paths followed by different water drops can be assumed as moving points traveling from one position to another in a multidimensional search space following different trajectories. RFD starts with the initialization of water drops to the initial node, *i.e.*, `initializeDrops()`. These nodes can be viewed as positions in a multidimensional search space and the altitudes of all nodes are initialized with higher values except destination nodes, which are set to zero for maintaining a downslope from start point to the end point (from origin to destination), i.e., `initializeNodes()`. The water drop from any node to the neighbors is traversed probabilistically [84] by following (4.8).

$$Pr(n_x, n_y) = \begin{cases} \frac{D_s(n_x,n_y)}{\sum_{k=1}^{\deg(n_x)} D_s(n_x,n_k)} & if\ n_y \in N(n_x), \\ 0 & if\ n_y \notin N(n_x), \end{cases} \tag{4.8}$$

where $D_s(n_x, n_y)$ denotes the decreasing gradient, and $D_s(n_x, n_y) = \frac{Alt(n_x)-Alt(n_y)}{w(n_x,n_y)}$; $\deg(n_x)$ represents the number of nodes connected to n_x; $Alt(n_x)$ and $Alt(n_y)$ represent altitudes of node n_x and n_y, respectively; $w(n_x, n_y)$ denotes weight of edge $e(n_x, n_y)$ and $N(n_x)$ denotes the set of neighboring nodes of node n_x. Further, continuous movement of water drops causes wearing away of soil and rocks, which leads to erosion (`erodePaths()`), *i.e.*, when a drop travels from n_x to n_y, $Alt(n_x)$ is eroded (reduced) and the new altitude, $Alt'(n_x)$ is defined as follows [85],

$$\begin{aligned} Alt'(n_x) &= Alt(n_x) - E(n_x, n_y), \\ E(n_x, n_y) &= \frac{\epsilon}{(\hat{N}-1).d}.D_s(n_x, n_y), \end{aligned} \tag{4.9}$$

where $E(n_x, n_y)$ denotes the amount of erosion when a drop travels from n_x to n_y; ϵ represents the erosion parameter; $\hat{N}$ denotes the number of nodes, and d is the number of drops used in the algorithm. When the erosion process is completed, the altitudes of nodes are marginally incremented by depositing sediments, *i.e.*, `depositSediments()`. In such cases, the amount of sediment deposition at any node n_k depends upon the amount of erosion in the

Algorithm 7: River Formation Dynamics (RFD) scheme [84]

```
initializeDrops()
initializeNodes()
while (not allDropsFollowTheSamePath() and not
otherEndingCondition()) do
  moveDrops()
  erodePaths()
  depositSediments()
  analyzePaths()
end while
```

previous step as follows [85],

$$Alt(n_k) = Alt(n_k) + \frac{\hat{E}}{\hat{N} - 1}, \tag{4.10}$$

where $\hat{E}$ denotes the sum of total erosion when a drop travels from origin to n_k. As excess erosion may develop flat surfaces across landscapes (the altitudes of nodes may become equal or close to zero) making the gradient evaluation insignificant, it is necessary to increase altitudes of the nodes by depositing sediments to avoid any premature convergence. In the end, all possible potential paths are analyzed to select the optimum one, $i.e.$, `analyzePaths()`.

4.3.2 Minimizing IR drop using RFD

Considering power distribution network as shown in Figure 4.1(b), IR drops (e.g., ϑ_x at node x) are evaluated by performing steady-state analysis before beginning of the optimization process. This evaluation provides information on the IR drop profile of entire PDN. The procedure for optimization of IR drop across each node of power distribution network is described in Algorithm 8 and the pseudocode for optimization of IR drop at a node x is described in Algorithm 9.

During optimization, the movement of water drops can be regarded as moving design variables (e.g., metal width ($w(x, x_{neighbor})$) and current sink (I_x)) towards convergence within a solution space. For starting the optimization process, a random population of design variables are generated and the drops are initialized to the design variables. The movement of drop

Algorithm 8: Procedure for minimization of IR drop using RFD method

1: Before the optimization process, the entire power distribution network is analyzed using KLU method to determine the IR drop profile of the network.
2: The IR drop of a node is minimized by employing RFD heuristic considering the widths and currents as design parameters.
3: The branch conductances connected to the node are updated along with node potential and current sink in each iteration.
4: All design constraints are evaluated and checked for any violations.
5: The process is continued for other affected nodes, $i.e.$, nodes having IR drop issues.
6: Finally, total IR drop and wire area of entire network are checked for any further violations or improvement and
the process is terminated if no further violation is observed.

Algorithm 9: Procedure for finding minimum IR drop for a node using RFD scheme

Input : A power distribution network, having a node x, which is connected to a number of neighbor nodes, such as $x_{neighbor}$, width ($w(x, x_{neighbor})$), IR drop (ϑ_x), current sink (I_x).

Output : minimum IR drop, ϑ

1: $total\_erosion \leftarrow \phi$
2: **while** *stopping criteria* **do**
3: $r \leftarrow$ `random(0,1)`
4: **for** $k \leftarrow 1$ to $num\_neighbors$ **do**
5: **if** $r < Pr(x, x_{neighbor})$ **then**
6: $break$
7: **end if**
8: **end for**
9: $x_{next} \leftarrow x^k$ ▷ find the next node
10: $total\_erosion \leftarrow total\_erosion + Erosion(x, x_{next})$ ▷ update erosion using (4.9)
11: **if** $\vartheta_x == \phi$ **then**
12: $\vartheta_x \leftarrow \vartheta_x + \epsilon$ ▷ $\epsilon \leftarrow (total\_erosion)/(N-1)$ denotes sedimentation parameter
13: **end if**
14: $\vartheta_x \leftarrow \vartheta_x - total\_erosion$
15: $x \leftarrow x_{next}$
16: **end while**

from one node to other node is estimated through the probability described in (4.8) and IR drop is reduced by the penalty being paid through the amount of erosion as described in (4.9). The subsequent values of design variables are updated in each iteration and the process is continued for a predefined number of iterations. Multiple experiments (i.e., 25 independent runs) are performed on each node. The mean of solutions is evaluated and is assigned the optimal IR drop at the node.

4.3.3 Minimizing metal area using RFD

Similarly, the process of minimizing metal area of the power distribution network starts with the analysis of the network to evaluate IR drop at each node. However, as each node in a network is connected to a number of other nodes in the PDN, altering any branch's metal width ($e.g.$, w_{xy}, where y=1, 2, 3, 4) affects IR drop profile of the connected nodes. Therefore, instead of analyzing a single branch at a time, we divide the network into several cells as shown in Fig. 4.1(c) and the wires in the cells are considered to have same width and length in order to reduce the number of optimization parameters. The overall optimization procedure is described in Algorithm 10. Here, the search strategy of RFD is utilized to pursue direction for finding optimum area of a PDN. The movement of water drops across the landscape can be

Algorithm 10: Procedure for minimization of area using both RFD method

1: Before the optimization process, the entire power distribution network is analyzed using KLU method to determine the IR drop profile of the network.
2: The area of a single cell is minimized by employing RFD heuristic considering the widths and currents as design parameters.
3: The branch conductances of the cell are updated along with node potential and current sink.
4: All design constraints are evaluated and checked for any violations.
5: The process is continued for other affected cells, $i.e.$, cells having IR drop issues.
6: Finally, total IR drop and wire area of entire network are checked for any further violations or improvement and
the process is terminated if no further violation is observed.

viewed as moving design variables (e.g., width (w) and current sink (I)) towards convergence within a solution space. To start the process of optimization, water drops are initialized to different network design variables and the subsequent values are updated by reinforcing the gradient calculations of each iteration. The probable values of design variables in the next iteration are evaluated by evaluating probability at each point of selection for all the design variables and design constraints. This probability is evaluated by using (4.8), where altitude values correspond to the node potentials (evaluated from steady state analysis) and the weight corresponds to the conductance between two nodes (say g_{xy} between node x and node y) of a cell. In this way, slopes and corresponding probabilities of all the nodes in a PDN are evaluated. Since conductance between two nodes changes during evaluation, corresponding

Algorithm 11: Procedure for finding the minimum width in a cell using RFD scheme

Input : Cells of a power distribution network, where each cell includes a node x, at least a neighbor node $x_{neighbor}$ and width between them ($w(x, x_{neighbor})$).

Output : width w

1: $total\_erosion \leftarrow \phi$
2: **while** *stopping criteria* **do**
3: $r \leftarrow$ `random(0,1)`
4: **for** $k \leftarrow 1$ to $num\_neighbors$ **do**
5: **if** $r < Pr(x, x_{neighbor})$ **then**
6: *break*
7: **end if**
8: **end for**
9: $x_{next} \leftarrow x^k$ ▷ find the next node
10: $total\_erosion \leftarrow total\_erosion + Erosion(x, x_{next})$ ▷ update erosion using (4.9)
11: **if** $w(x, x_{next}) == \phi$ **then**
12: $w(x, x_{next}) \leftarrow w(x, x_{next}) + \epsilon$
▷ $\epsilon \leftarrow (total\_erosion)/(N-1)$ denotes sedimentation parameter
13: **end if**
14: $w(x, x_{next}) \leftarrow w(x, x_{next}) - total\_erosion$
15: $x \leftarrow x_{next}$
16: **end while**

width, w_{xy} between node x and node y, also changes. The neighboring node having maximum probability is selected as the next point in the path of water drop and the width is reduced by an amount of erosion evaluated using (4.9). The constraints are checked for any violation in each iteration using constraint dominance principle [7]. The procedure for evaluating optimal width of a cell using a single drop is described in Algorithm 11. Further, as excess erosion may lead to termination of optimization process and convergence to the local optimum solution. In order to avoid falling into local traps (local optimum solution) due to excess erosion, probable values of design parameters (node potentials and widths) are updated using (4.10). The process is terminated after a predefined number of iterations (stopping criteria). Accordingly, a number of drops are employed to follow different paths from the starting nodes, which results in many possible solutions. Finally, all possible solutions are analyzed and optimum solution is evaluated by selecting the best fitness value. However, instead of selecting the final solution of a single experiment, 25 independent runs are performed each time on a cell and the mean of all solutions is represented as the final optimum solution.

During minimization process (both IR drop and metal area), the boundary constraints are handled using random boundary constraint handling strategy [115] and the function constraints are handled using constrained-dominance principle [7]. The details on the constraint handling strategies are described in chapter 5.

4.3.4 Limitations of RFD

River formation dynamics accounts for the maximum possible combinations of proximal paths to take advantage of locality and to facilitate flow of the drops in the search space. Since flow of the drops is governed by the probability in (4.8), the explosion of search space is random and the possibility of encountering a flat surface during mid-flow is quiet conventional. In addition to that, the random amount of erosion in the altitude of traversed nodes in G results in uncertainties of the drop's performance index. This uncertainty is limited by adding arbitrary sediment values to the altitude, which can induce drops to follow a flat surface infinitely many times and subsequently diverging flow of the drop.

4.3.5 River Formation Dynamics Revisited with Proposed New Developments

Even though the current river formation dynamics scheme is capable of determining an optimal solution, when compared with other evolutionary methods, the ability to find a global optimum solution in all the scenarios is comparatively weak. The reason for such weak performance is due to the lack of sufficient tuning of parameters (e.g., longitudinal slope, stochastic flow) at the end of the search and also the lack of efficient utilization of certain key factors, such as transverse slope, sediment transport rate, etc. during analysis. Therefore, better realization of water flow is necessary to find an optimum solution accurately and efficiently. Considering it, we propose to improve the performance of RFD by incorporating a new stochastic flow model based on sediment transport rate affected by transverse slope characteristics.

4.3.6 Proposed Model Equation

Similar to the Brownian motion of water flow, we consider the search toward optimum solution in RFD to be guided by a stochastic variable model. In view of this, we formulate a model centered around the volume of water flowing at certain instant of time. As the volume of water flow is often affected by the amount of sediment being transported, we consider volume of the sediment deposited at the bottom being transported per unit width per unit time as a decision variable during formulation of model equation. At first, we establish the equation governing transverse (cross) slope and flow velocity (u) in two different directions, i.e., downstream (s) and transverse (n), while considering the case of downstream elevated channel bed. A channel bed is shown in Figure 4.2 having mild transverse slope (S_n) with uniform sized sediment materials. In this regard, volume of sediment transport rate per unit width per unit time in both s and n directions (i.e, q_s and q_n, respectively) can be related to flow velocity using (4.11) [116],

$$\frac{q_n}{q_s} = \frac{u_n}{u_s} + \beta S_n, \tag{4.11}$$

where, $\beta \in [1, 2]$ is a dimensionless parameter. As water drops choose to flow in different paths, both longitudinal and transverse slopes play a major role in making a decision, i.e., choosing a path. If flow varies in both s and n directions, probability of the water drop traversing in a particular direction described described in (4.8) can be rewritten as (4.12),

$$Pr(x_1, x_2) = \begin{cases} \frac{S_f(x_1,x_2)}{\sum_{x_k=1}^{\deg(x_i)} S_f(x_i,x_k)} & if\ x_j \in N(x_i), \\ 0 & if\ x_j \notin N(x_i), \end{cases} \tag{4.12}$$

where, S_f denotes the resultant slope and $|S_f| = \sqrt{S_l^2 + S_n^2}$; S_l represents longitudinal slope in s direction and, $S_l(x_1, x_2) = \frac{h_2 - h_1}{x_2 - x_1}$ and $S_n(x_1, x_2) = \frac{h_4 - h_2}{x_3 - x_1}$. As the effect of change in q_n to the flow of water drops is minimal, q_s can be considered as a decision variable for function evaluation during optimization. Further, with drops traveling in direction of decreasing altitude values, both slopes S_l and S_n start to decline as drops tend to reach flat surface (sea). In this regard, the volume of sediment transport rate per unit width per unit time in downstream

direction, q_s, can be considered to be minimum at sea level. Therefore, we consider q_s as a decision variable to be updated in each time step for a new function evaluation. As q_s is changed in each time step, it can be modeled by (4.13),

$$q_s = q_s + \delta q_s, \tag{4.13}$$

where, δq_s denotes the change in volume of sediment transport rate per unit width per unit time in s direction. To demonstrate the existence of any trend in (4.13), a parametric study is performed in section 4.3.7, which supports the effectiveness of the model described in (4.13). When a drop chooses certain probability Pr to follow a path, q_s changes. Therefore, a weighted random positive seed value, ε is added to q_s in each iteration as a token for choosing a better path probabilistically as described in Algorithm 12. Further, for each decision variable (say q_s), there exists a separate ε value which changes in each iteration. As Υ, where $\varepsilon \in \Upsilon$, is evaluated for all decision variables (for all q_s), which choose paths probabilistically to move in descent or ascent direction, it can be considered as a random vector in a multidimensional space spanned by the decision variables. Therefore, it is necessary to estimate the weighted mean vector, $\boldsymbol{\mu_\varepsilon}$ and weighted covariance matrix, $\boldsymbol{\Sigma_\varepsilon}$ for suitable estimation of ε for each decision variable.

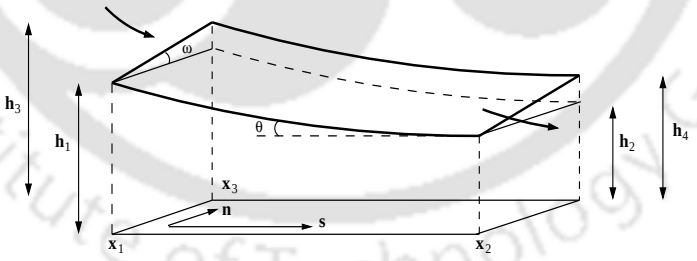


Figure 4.2: A slice of length $dx = (x_1 - x_2)$ along a river for the formulation longitudinal and transverse slope. The notations are: water depths at different locations h_i, where $i = 1, 2, 3, 4$, longitudinal slope angle θ, transverse slope w, downstream direction s, normal direction n.

Further, if the drop discovers an improved pattern (minimum sediment transport rate) in an iteration, the corresponding q_s value is stored and represented as regional sediment deposition rate per unit time per unit width, q_s^{rsed} of that iteration. The weighted difference between q_s^{rsed} and the current q_s value is added stochastically to the current q_s value to search for

Algorithm 12: Procedure for evaluation of ε

for $i \longleftarrow 1$ *to* $popSize$ **do**
  for $j \longleftarrow 1$ *to* $numVar$ **do**
    $c_{ij} \longleftarrow \phi$
    /* Sf denotes final slope */
    $Sf_{ij} \longleftarrow \sqrt{S_l^2 + S_n^2}$
    /* N denotes no. of possible paths */
    $Pr_{ij} \longleftarrow (Sf_{ij})/(\sum_{x=1}^{N} Sf_{ij})$
    Generate random number r_1 between 0 and 1
    if $r_1 < Pr_{ij}$ **then**
      $c_{ij} \longleftarrow c_{ij} + 1$
    end
    $c_{ij} \longleftarrow c_{ij} - r_1 \times c_{ij}$
  end
end
/* μ_ε denotes weighted mean vector of ε */
$\mu_\varepsilon \longleftarrow \frac{\sum_{i=1}^{popSize}\sum_{j=1}^{numVar} c_{ij} Pr_{ij}}{\sum_{i=1}^{popSize}\sum_{j=1}^{numVar} c_{ij}}$
/* Σ_ε denotes weighted covariance matrix of ε */
$\Sigma_\varepsilon \longleftarrow \frac{\sum_{i=1}^{popSize}\sum_{j=1}^{numVar} c_{ij}(Pr_{ij}-\mu_\varepsilon)^T(Pr_{ij}-\mu_\varepsilon)}{\sum_{i=1}^{popSize}\sum_{j=1}^{numVar} c_{ij}}$

an optimum solution in the regional direction. Moreover, with more number of water drops flowing in different directions, the weighted difference between best pattern found among all the drops, q_s^{lsed} (large scale sediment deposition) and individual current q_s value is also added to the current q_s value. These additions to the sediment's movement motivate the drops to search in two dimensional space. Considering these additions, change in volume of sediment transport rate per unit width per unit time, δq_s in (4.13) can be expressed as (4.14),

$$\delta q_s = \varepsilon + w_1 \times (q_s^{rsed} - q_s) + w_2 \times (q_s^{lsed} - q_s), \tag{4.14}$$

where w_1 and w_2 denote random positive coefficients. Random weighting of control parameters (q_s^{rsed} and q_s^{lsed}) in (4.14) can induce spurious trend during movement of sediments in (4.13) and can lead to explosion, similar to "drunkard's walk" of control parameters in traditional stochastic algorithms [68, 117]. It is shown in [68, 117] that the random variation of these positive coefficients can be constricted to provide a search domain for stochastic processes. However, these analyses rely on the assumption that the time series are stationary, i.e., the covariances or means of local best points are assumed constant to evaluate a stable point

for coefficients, e.g., w_1 and w_2. As random values of w_1 and w_2 can cause explosion [117], the occurrence of such adverse situation during optimization analysis renders the stationarity assumption inefficient [68]. Moreover, this assumption of stationarity during evaluation of certain control parameters with few other conditions during analysis do not provide much solace to experimental analysis of stochastic model equation described in (4.13). Therefore, in this paper, we consider to maintain a scheduled bound for both w_1 and w_2 coefficients through stability analysis [117].

4.3.7 Trend in Sediment Transport Rate

In this section, we perform likelihood ratio test (LRT) [118] to search for any spurious trend in the increments of sediment transport rate. For trended sediment transport, there exists a direction on the increments, while the expectation of increments is found to be zero for non-trended transport. It is necessary to explore random changes in the trajectory of sediment transport rate to discriminate a trended phenomenon from the non-trended one. Assuming existence of trend during sediment transport, (4.13) can be expressed as (4.15),

$$q_s^t = \eta + \rho q_s^{t-1} + \delta q_s^t, \quad (t = 2, 3, \ldots, n), \tag{4.15}$$

where, q_s^t denotes the sediment transport rate per unit width per unit time at time instant t; η denotes amount of drift and ρ denotes drift coefficient. For accounting any trend in the time series implementation of decision variable equation in (4.15), both independent and identically distributed (IID) increments are assumed along with known distribution of any trend in q_s at each time step. Further, LRT can be employed to perform a parametric test for η and investigate the hypothesis that $(\eta, \rho) \in [0, 1]$, assuming normally distributed increments to showcase a discretized movement in q_s. Considering null (H_0) and alternative hypotheses (H_1) for increment in q_s, log-LRT statistic, D can be evaluated using (4.16) [118],

$$D = -2ln(L(H_0)) + 2ln(L(H_1)), \tag{4.16}$$

where $L(H_0)$ and $L(H_1)$ are maximum likelihoods under H_0 and H_1 respectively, and both H_0 and H_1 can be expressed as (4.17),

$$H_0 : \; q_s^t = q_s^{t-1} + \delta q_s^t, \quad \delta q_s \overset{\text{IID}}{\sim} N(0, \sigma^2),$$
$$H_1 : \; q_s^t = \eta + \rho q_s^{t-1} + \delta q_s^t, \quad \delta q_s \overset{\text{IID}}{\sim} N(0, \sigma^2),$$

where δq_s is distributed with mean μ and variance σ^2.

Further, both η and ρ can be estimated by maximum likelihood estimators as (4.17),

$$\hat{\eta} = \mu_{(0)} - \hat{\rho}\mu_{(-1)}, \;\; \hat{\rho} = \frac{\sum_{t=2}^{n} (q_s^t - \mu_{(0)}) \sum_{t=2}^{n} (q_s^{t-1} - \mu_{(-1)})}{\left[\sum_{t=2}^{n} (q_s^{t-1} - \mu_{(-1)})\right]^2},$$

where $\mu_{(i)} = (n-1)^{-1} \sum_{t=2}^{n} q_s^{(t+i)}$, $i = -1, 0$, and both $\hat{\eta}$ and $\hat{\rho}$ denote least square estimators of η and ρ respectively. To investigate the distribution of both η in both null and alternate hypotheses, an analogy is presented to regression t-statistic [118] for measuring $\hat{\eta}$. The equivalent t-statistic for $\hat{\eta}$ can be represented as (4.17),

$$\hat{\tau}_\eta = \frac{\hat{\eta}}{\sqrt{c_1 S_e^2}}, \tag{4.17}$$

where S_e^2 is the regression residual mean square and $S_e^2 = (n-k-1)^{-1}(\mathbf{Q}^T(\mathbf{I} - \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T)\mathbf{Q})$, $\mathbf{Q} = [q_s^1 \; q_s^2 \ldots q_s^{t-1} \; q_s^t \ldots q_s^n]$; $\mathbf{I}$ represents identity matrix; $\mathbf{X}$ represents $(n-1) \times k$ matrix of independent regression variables from right hand side of the appropriate regression equation and c_1 is an element of $\mathbf{X}^T\mathbf{X}$ associated with $\hat{\rho}$.

To study the characteristics of trend in sediment transport rate q_s at any time instant t, we perform empirical analysis of $\hat{\tau}_\eta$ on alternate hypothesis model H_1. Both t-value and p-value are evaluated for statistical analysis of finite samples generated using H_1 hypothesis model. Twelve replicas of $20,000$ samples are generated by considering several combinations of three η $(0.001, 0.01, 0.5)$ values and four ρ $(1, 0.5, 0.01, 0.001)$ values. Two different t-statistics are employed by considering different samples to observe the significance level (p-value) and results of the statistic test are listed in Table 4.1. First statistic test is performed independently

Table 4.1: Empirical analysis of $\hat{\tau}_\eta$ for $(\eta, \rho) \in (0, 1)$ in $q_s^t = \eta + \rho q_s^{t-1} + \delta q_s^t$.

ρ	η	t-value	df	p-value
1	0.001	3.3982 (1.6870)	667 (832)	3.5930e-04 (0.0460)
	0.01	3.2934 (2.4342)	337 (358)	5.4760e-04 (0.0077)
	0.5	3.1259 (2.7524)	147 (149)	0.0011 (0.0033)
0.5	0.001	3.4106 (0.0670)	300 (609)	3.6850e-04 (0.2516)
	0.01	3.2563 (0.6390)	222 (433)	6.5271e-04 (0.2616)
	0.5	3.1039 (0.2706)	317 (626)	0.0010 (0.3934)
0.01	0.001	3.1914 (0.2408)	169 (351)	8.4410e-04 (0.4049)
	0.01	3.1606 (0.5162)	147 (298)	9.5609e-04 (0.3030)
	0.5	3.1496 (0.7437)	131 (251)	0.0010 (0.2289)
0.001	0.001	3.1655 (0.1412)	171 (353)	9.1650e-04 (0.4439)
	0.01	3.1745 (0.2649)	162 (336)	8.9830e-04 (0.3956)
	0.5	3.1462 (0.7364)	128 (245)	0.0010 (0.2311)

on each sample of alternate hypothesis model only, and second test is performed by comparing samples of both null and alternate hypothesis models (paired t-statistic listed within braces in Table 4.1) [118]. Means of two populations are kept equal during second statistic test. It can be observed from Table 4.1 that for paired t-statistic, three replicas are below $0.05(5\%)$ significance level and other replicas are between above 0.05 significance level. In view of this, it is accounted that the estimated null and alternate hypothesis models are valid when ρ is kept at 1.

Further, it can be observed from Figure 4.3 that with decrease in η value keeping ρ fixed at 1, sample values generated by following H_1 hypothesis model converges toward the sample values of H_0 hypothesis model. On other hand, for first t-statistic, all replicas are observed to be below $0.01(1\%)$ significance level, which explains less occurrence of randomness in the alternate hypothesis model H_1. The regression t-statistic test values (both t-values and p-values) listed in Table 4.1 establish a correlation between trended and original flow of sediments at certain instant of time. It can be emphasized that IID increments are assumed in δq_s for generating samples of H_0 hypothesis model for investigating the characteristics of trend in q_s. However, uncorrelated increments occur during sediment transport, and the statistic test performed for investigating trend in the sediment transport rate gives little insight into any variation in the behavior of increments. Therefore, it is necessary to perform a parametric

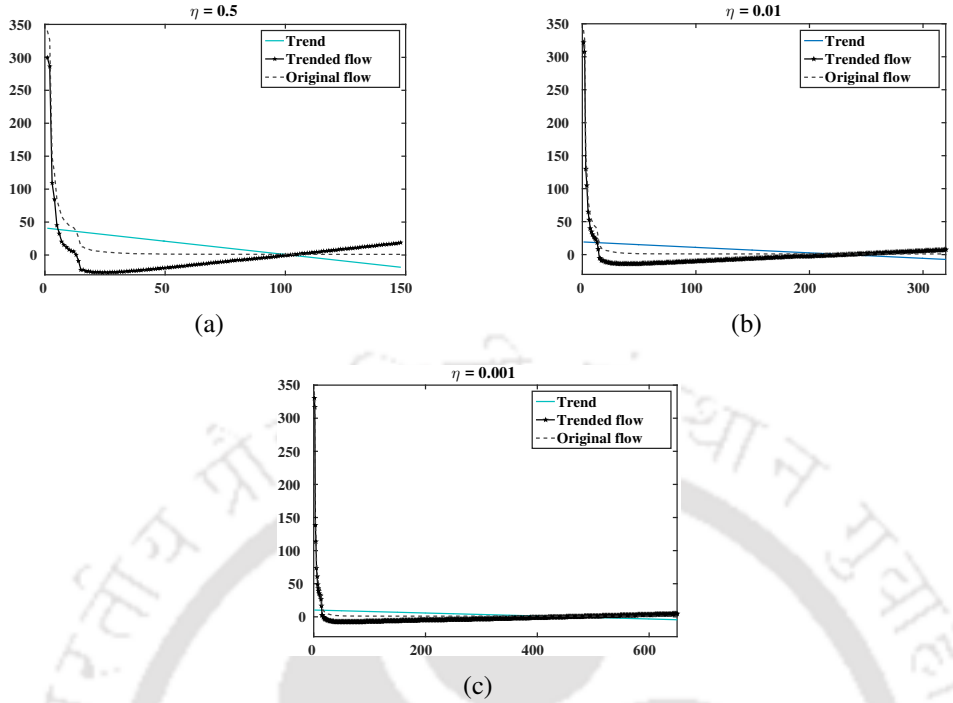


Figure 4.3: (a) Plots of global fitness values of Rosenbrock test function at different trend amount, (a) $\eta = 0.5$, (b) $\eta = 0.01$, (c) $\eta = 0.001$.

study through stability analysis for exploring the impact of variations in the increments by establishing a bound on coefficients (w_1 and w_2).

4.3.8 Stability Analysis

When a random search procedure finds an optimum solution, its probability can be estimated as, $\lim_{m\to\infty} P\{\theta_i \subset \Theta\} = 1$, where Θ is a set of global optimal solutions to optimization problems and θ_i is the estimate of optimal solution after i iterations. The search procedure becomes progressively more conservative with increase in the number of iterations in order to constrain the movement of points which move away from promising positions. This affects convergence of the search procedure unless a strong evidence is put forward for any improvement in the result. Although a set of random search procedures follow a number of trials[2] per iteration as the iteration grows, convergence is only guaranteed asymptotically for infinite number of iterations [117]. For avoiding the need to force a random search procedure to become progressively more conservative in obtaining convergence, the method presented

[2]Here, trial denotes a procedure which is repeated many times and has a set of possible outcomes.

in [119] proposes to aggressively follow a set of promising points frequently visited by the estimate θ_i as an estimate of optimal solution. These procedures follow a irreducible, time homogeneous and positive recurrent Markov chain events, which generate a sequence of estimates θ_i that fail to converge in most of the cases (the cases that require conservative approach towards progress of finding global optimal solution). Many random search procedures including the methods discussed in this section can be improved by making appropriate changes in the optimal solution estimation procedure. In particular, rather using current point or the most frequently visited point to estimate the optimal solution, progress in estimates can be guided by finding the best estimates in local search space and among all estimates [68]. The approach of using a point (either global or local best) with the best estimated objective function value to search for optimal solution is known for a long time. This type of search is employed to analyze stability of the algorithm and behavior of coefficients in the algorithm to ensure convergence. However, to prevent any explosion and for suitable estimation of coefficients in RFD, we investigate the behavior of these coefficients on δq_s by considering three different cases.

(i) If the sequence of generated variables are close to the regional territory, (4.14) can be rewritten as (4.18),

$$\delta q_s = \varepsilon + a_1 \times (q_s^{rsed} - q_s), \tag{4.18}$$

(ii) If the sequence of generated variables are close to the large scale territory, (4.14) can be rewritten as (4.19),

$$\delta q_s = \varepsilon + a_2 \times (q_s^{lsed} - q_s), \tag{4.19}$$

(iii) If the sequence of generated variables are in between regional and large scale territories, (4.14) can be rewritten as (4.20),

$$\delta q_s = \varepsilon + a_3 \times (\mu_s - \sigma_s - q_s) + a_4 \times (\mu_s + \sigma_s - q_s), \tag{4.20}$$

where μ_s and σ_s denote mean and standard deviation of uniformly distributed points between q_s^{rsed} and q_s^{lsed}, respectively.

As the type of convergence for an iterative stochastic optimization algorithm, like RFD, is usually investigated through considering all possible sequences of variables generated during evaluation process, (4.14) can be rewritten as a combination of above three cases as (4.21),

$$\begin{aligned}
\delta q_s &= \varepsilon + a_1 \times (q_s^{rsed} - q_s) + a_2 \times (q_s^{lsed} - q_s) + a_3 \times (\mu_s - \sigma_s - q_s) + a_4 \times (\mu_s + \sigma_s - q_s) \\
&= \varepsilon + a_1 \times (q_s^{lbest} - q_s) + a_2 \times (q_s^{gbest} - q_s) + a_3 \times \left(\tfrac{q_s^{lbest}+q_s^{gbest}}{2} - \sigma_s - q_s\right) \\
&\quad + a_4 \times \left(\tfrac{q_s^{lbest}+q_s^{gbest}}{2} + \sigma_s - q_s\right) \\
&= \varepsilon + \left(a_1 + \tfrac{a_3+a_4}{2}\right) \times (q_s^{lbest} - q_s) + \left(a_2 + \tfrac{a_3+a_4}{2}\right) \times (q_s^{gbest} - q_s) + \sigma \times (a_4 - a_3) \\
&\approx \varepsilon + w_1 \times (q_s^{rsed} - q_s) + w_2 \times (q_s^{lsed} - q_s) + \sigma \times w_3
\end{aligned} \tag{4.21}$$

Equation (4.21) involves a system of having three coefficients (w_1,w_2,w_3). The system can be simplified further by considering, $w = w_i, i \in [1,3], i \in N$. If the feasible region of operation (say $\mathbb{R}$) is considered to be finite, the system would contain sum of all the estimates of the generated variables. In view of this, the basic simplified system at any instant t can be defined by considering all three cases of generated variables as (4.22) (assume $w_i \equiv a_i$ for simplicity),

$$\begin{aligned}
y^{t+1} &= (1-w)q_s^t + wq_s^{rsed}, \\
y^{t+1} &= (1-w)q_s^t + wq_s^{lsed}, \\
y^{t+1} &= (1-2w)q_s^t + w(q_s^{rsed} + q_s^{lsed}),
\end{aligned} \tag{4.22}$$

where $y^t = q_s^t - \varepsilon$. Considering $\mathbf{Q}_s^t = \begin{bmatrix} q_s^t & q_s^{rsed} & q_s^{lsed} \end{bmatrix}^T$, sediment rate at any instant t in $\mathbb{R}^3$ and, coefficient matrix,

$$\mathbf{W} = \begin{bmatrix} 1-w & w & 0 \\ 1-w & 0 & w \\ 1-2w & w & w \end{bmatrix},$$

then system can be realized through $\mathbf{Y}^{t+1} = \mathbf{W}\mathbf{Q}_s^t$. Following Cardano's formula [120], the eigenvalues of $\mathbf{W}$ can be represented as (4.23),

$$\begin{cases} \lambda_1 = & \frac{1}{3}(1+2w) + (S+T), \\ \lambda_2 = & \frac{1}{3}(1+2w) - \frac{1}{2}(S+T) + \frac{1}{2}i\sqrt{3}(S-T), \\ \lambda_3 = & \frac{1}{3}(1+2w) - \frac{1}{2}(S+T) - \frac{1}{2}i\sqrt{3}(S-T), \end{cases} \tag{4.23}$$

where $S = (R+\sqrt{V^3+R^2})^{1/3}$, $T = (R-\sqrt{V^3+R^2})^{1/3}$, $R = -\frac{1}{27}(13w^2+4w+1)(1+2w)$ and $V = -\frac{1}{9}(w^2+4w+1)$. Further, we can define a diagonal matrix $\mathbf{\Lambda}$ with non-zero eigenvalues such that,

$$\mathbf{\Lambda W \Lambda}^{-1} = \mathbf{\Psi} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}. \tag{4.24}$$

In view of this, we can illustrate, $\mathbf{Y}^{t+1} = \mathbf{\Lambda}^{-1}\mathbf{\Psi}\mathbf{\Lambda}\mathbf{Q}_s^t$. In particular, oscillation in the system will occur, if and only if,$\mathbf{Y}^{t+1} = \mathbf{Q}_s^t$ for $\mathbf{\Psi} = \mathbf{I}$, where $\mathbf{I}$ denotes an identity matrix. Although (4.23) represents three different eigenvalues $(\lambda_1, \lambda_2, \lambda_3)$, it provides little information regarding their characteristics. However, it can be accomplished by evaluating different aspects of polynomial discriminant D, where $D = (V^3 + R^2)$.

4.3.8.1 Case I

For $D > 0$, one eigenvalue (λ_1) is real and two (both λ_2 and λ_3) are complex conjugates. More precisely, for $D = 1$, the complex eigenvalues can be represented as, $\lambda_2 = cos\theta + isin\theta, \lambda_3 = cos\theta - isin\theta$, where $cos\theta = \frac{1}{3}(1+2w) - \frac{1}{2}(R+1)^{1/3}$ and $sin\theta = \frac{\sqrt{3}}{2}\left[(R+1)^{1/3} - (R-1)^{1/3}\right]$. Further, considering $R = 1$ and $V = 0$, the constriction coefficient can be expressed as, $w = 0.4 + (1.5 \times cos\theta)$. In this regard, different nontrivial values of $0.4 \leq w < 2$ can be evaluated at different values of θ. This sets a strict upper bound for constriction coefficient w. It is noted that for evaluating a stable upper and lower bound, negative values of w are neglected [68]. In addition, other combinations of R and V are overlooked for negative values of w or $w \in [0.4, 2)$.

4.3.8.2 Case II

For $D = 0$ (both $R = 0$ and $V = 0$), all eigenvalues are real, where at least two eigenvalues are found to be equal. This condition generates negative values of w for $R = 0$ (i.e.,$-1/27(13w^2 + 4w + 1)(1 + 2w) = 0$) and $V = 0$ (i.e.,$-1/9(w^2 + 4w + 1) = 0$).

4.3.8.3 Case III

For $D < 0$, all eigenvalues are found to be real and unequal. Further, for $D < 0$, $V < 0$ and $|V| > R$, considering $V = -1$ and $R = 0$, $-\frac{1}{9}(w^2 + 4w + 1) = -1$ condition can be employed to evaluate a stable bound for w. Solving this, a fixed lower bound of $w = 1.464$ can be obtained while evaluating a scheduled bound.

We conclude this analysis by summarizing that an estimate of constriction coefficient, i.e., $1 < w < 2$ can be used as an estimate of evaluating the points with the best estimated objective function values[3]. Although the proposed developments in the model described in section 4.3.6 follow a random probabilistic search, a scheduled bound for constriction coefficients (w_1, w_2, w_3) in (4.21) supports the direction of return of estimated objective function value at the estimated optimal solution in each iteration. This favors the iteration to converge faster for such a random procedure.

4.3.9 Performance on Single Objective Test Functions

In this section, we provide simulation results of proposed modified RFD (MRFD) method after optimizing a set of single objective test problems. Here, the set of single objective test problems used as benchmark and the metric applied for evaluating quality of solutions are described. Finally, we evaluate our proposed method and compare the performance with other standard optimization algorithms. The proposed modified RFD is implemented by using C/C++ language and all simulation experiments are performed on Linux environment on a machine with Intel Xeon E5-2620 processor having 64GB of RAM.

[3]Please refer Appendix B for evaluation of lower bound of w.

4.3.9.1 Test Problems

In this section, we provide insight into different sets of both constrained and unconstrained problems solved during analysis. These problems have been used in the literature as benchmarks. We have selected 35 different single objective problems for evaluation, in which 23 are unconstrained and 12 are constrained test problems. The first set of 18 unconstrained test functions[4] are based on different genre and five set of unconstrained scalable test problems ($F1$ to $F5$) are selected from CEC-2014 single objective benchmark suite (having dimension[5] 10) [121]. The second set of 12 constrained problems consist of seven constrained test functions of different genre[4], i.e., $Rosenbrock1$, $Rosenbrock2$, $Mishra$, $Simionescu$, $g11$, $g12$ and $g13$, and five constrained test problems ($C01$ to $C05$) of dimension 10. These five constrained test problems ($C01$ to $C05$) are selected from CEC-2017 single objective benchmark suite [122]. The basic properties of all test functions are listed in Table 4.2. All benchmark functions can also be categorized into unimodal and multimodal with increase in test function dimension. For unimodal functions, the convergence rates are crucial as optimum solutions of these functions can be obtained without much difficulty. On the other hand, solving multimodal functions reflect the ability of MRFD in avoiding local optima and converging to desired near-global solution.

4.3.9.2 Peer algorithms and parameter settings

With aim of giving a complete overview on the performance of proposed MRFD, four different state-of-art optimization techniques, i.e., SPSO [67], G-CMA-ES [123], UMOEAII [124] and L-SHADE [65, 66] are considered for evaluation on single objective test problems. For the purpose of comparison, we have used the authors implementation of all these algorithms available in public domain.

Standard particle swarm optimization (SPSO) algorithm presented in [67] is used as a

[4]These are standard single objective optimization test functions used in literature. They are also available at https://en.wikipedia.org/wiki/Test_functions_for_optimization and at http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO_files/Page422.htm

[5]'Dimension' means number of decision variables.

baseline for performance testing and to represent PSO in literature. During analysis using SPSO, constriction factor (χ) is set to 0.72984, and constriction coefficients (c_1 and c_1) are kept at 2.05 each to ensure convergence [125]. G-CMA-ES[6] is a variation of Covariant Matrix Evolutionary Strategy (CMA-ES) that avoids premature convergence by restart strategy, which doubles the population size on each restart and search for global solution in the increased search space. We have considered the values for different parameters of G-CMA-ES as suggested by Auger and Hansen [123]. The initial solution in G-CMA-ES is chosen uniformly from the domain and the initial distribution size is set to one third of domain size [123]. UMOEAII[7] is a united multi-operator evolutionary algorithm framework, which incorporates multi-operator differential evolution (DE) algorithm, multi-operator genetic algorithm (GA) and CMA-ES algorithm. The different parameters used during implementation including the respective parameter values are considered according to the authors implementation as described in [124]. Further, L-SHADE algorithm is a variant of SHADE algorithm, which uses a strategy to linearly reduce the size of population in each generation. L-SHADE algorithm is used to solve unconstrained test functions considered in this thesis and an enhanced version of L-SHADE (i.e., L-SHADE44) algorithm is considered during evaluation of constrained test functions only. It should be noted that L-SHADE44 is the winner of CEC-2017 competition, which is performed on single objective constrained real parameter optimization problems. The different parameters used during implementation of both versions of L-SHADE algorithm are considered according to the authors implementation described in [65, 66].

As the choice of population size plays a prime factor in performance of algorithms, a minimum population size of 100 is considered during evaluation of all these algorithms. Further, to assess the performance of algorithms on single objective test problems quantitatively, success performance (SP or FEs) is used as a measure for the expected number of function evaluations (FEs) to reach a target function value presented in [126]. SP is an empirical measure on the ability to generalize performance results of algorithms and it can be evaluated

[6]G-CMA-ES is the winner of CEC-2005 competition on single objective real parameter optimization.

[7]UMOEAII is the winner of CEC-2016 competition, which is performed on CEC-2014 single objective benchmark suite.

as [126],

$$SP = f_{mean} \times \frac{Run_{total}}{Run_{success}}, \tag{4.25}$$

where f_{mean} denotes mean of successful runs achieved by the algorithm; Run_{total} and $Run_{success}$ denote number of total runs and number of successful runs achieved by the algorithm, respectively.

4.3.10 Performance Assessment

During implementation of MRFD, a proper range of key parameters is considered. The coefficients are assigned reasonably to ensure stability and convergence, and to attain global solution. A judgment on this is reached through stability analysis described in section 4.3.8. By adjusting the value of constriction coefficient w (= (w_1, w_2, w_3)), the sediment rate has a greater tendency to constrict itself within the area having best fitness and explore nearby region for global solution. In the proposed method, it is possible to weight the sediment rate to favor the best fitness evaluation by varying the coefficients w_1, w_2 and w_3. Different values of these coefficients are generated randomly in each iteration between $(1, 2)$ for better exploration of alternate areas in the search space to achieve global optimum. The probability seed value ε is generated probabilistically according to Algorithm 12 for a guided search towards minimum sediment transport rate. The control of coefficients on sediment transport rate, similar to the role of friction or shear stress (or erosion) on the movement of sediments in flow direction, has resulted in the improved performance of MRFD. The performance of proposed MRFD is evaluated based on the optimization results on 35 test functions as compared with peer algorithms. For each test function, 25 independent runs are performed to observe the success in function evaluation. Table 4.2 presents the optimization results obtained by proposed MRFD in comparison with peer algorithms. It also summarizes the minimum FEs taken by algorithm to evaluate each test function including number of successful function evaluations for all the algorithms. The success rate (mentioned in parenthesis in Table 4.2) and the FEs of each algorithm is evaluated using SP metric, which serves as a measure of computational

effort (i.e., number of FEs). For algorithms, which fail to converge to a global optimum solution (number of FEs is not available), rank of median final function value is obtained and listed within angle brackets in Table 4.2. It can be observed that MRFD is able to locate near-optimal solution for all 35 test functions with relatively better success rate and minimum FEs, indicating effectiveness and statistical stability. During numerical analysis, mean of best fitness is considered out of 25 runs to observe the evolution in the movement of sediments and the standard deviation of best fitness is evaluated to observe the consistency in achieving global optimum solution.

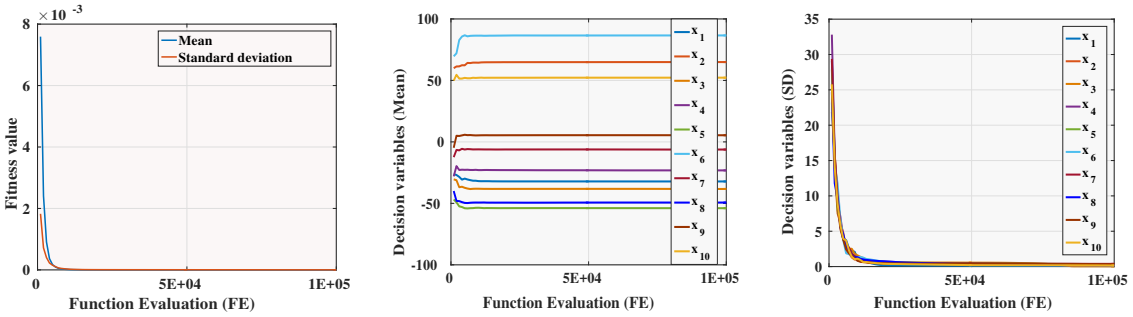
It can be observed from Table 4.2 that proposed MRFD method demonstrates sufficient improvement in solving these test functions with minimum number of FEs over traditional RFD method in solving unimodal functions (i.e., $F1$-$F5$, $Sphere$ and $Rosenbrock$). The success rate for completing 25 independent runs is 100% in all the cases. The MRFD takes $1e+05$, $1.5e+05$, $1e+05$, $1.1e+05$ and $1e+05$ function evaluations for $F1$, $F2$, $F3$, $F4$ and $F5$ test functions, respectively, to achieve convergence using ten decision variables. However, except for $F2$ and $F4$ test functions, MRFD demonstrates superior performance over SPSO and G-CMA-ES in solving unimodal test functions in terms of number of FEs. UMOEAII shows better performance for $F2$ test function and L-SHADE demonstrates better performance while solving $F4$ test function. On the other hand, for functions, $Rosenbrock$ and $Sphere$, the results listed in Table 4.2 show a competitive performance of MRFD as compared to all peer algorithms. Both $Sphere$ and $Rosenbrock$ functions are implemented for five decision variables (dimension=5). It takes 10400 and 100 function evaluations for $Rosenbrock$ and $Sphere$ functions, respectively, to achieve satisfactory accuracy. To demonstrate the effectiveness of MRFD over unimodal functions, plots of mean and standard deviations of fitness values and decision variables over 25 runs are shown in Figure 4.4 for $F3$ function. It can be observed from Figure 4.4(a) that MRFD quickly converges to global optimum solution within a relatively small number of FEs ($< 1e + 05$) for $F3$ function. However, the decision variables converge at $1e+05$ FEs (refer Figure 4.4(b) and Figure 4.4(c)). Therefore, minimum

FEs are evaluated after verification of convergence of all decision variables in 25 independent runs.

Table 4.2: Performance analysis of MRFD on different single objective test functions.

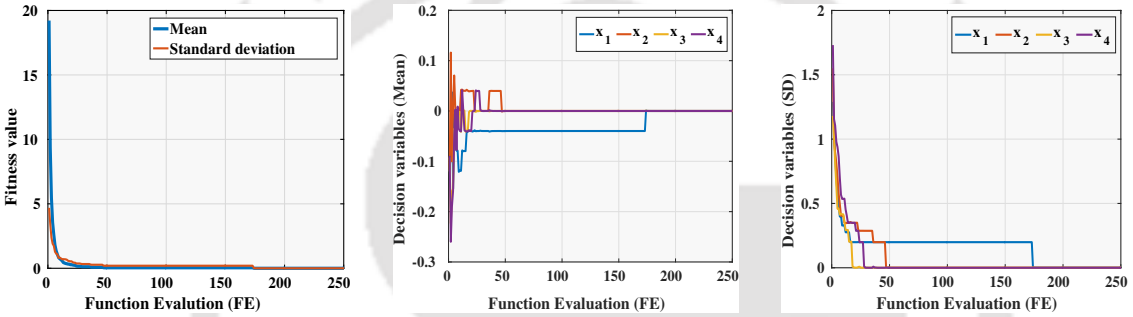
Test Functions	*SPSO*	*UMOEAII*	*LSHADE*	*G-CMA-ES*	*RFD*	*MRFD*	min. FEs	Cost Function Property/Shape/Number of constraints
Unconstrained Functions								
3-hump camel	1(100%)	1(100%)	1.2(100%)	8(100%)	4(100%)	1(100%)	100	Multimodal, three local minima, valley-shaped, distributed
Ackley	1(100%)	1(100%)	1(100%)	8.5(100%)	3(80%)	1(100%)	210	Multimodal
Beale	1(100%)	1(100%)	1(100%)	⟨6⟩	4(100%)	1(100%)	100	Multimodal, sharp peaks at corner
Booth	1(100%)	1(100%)	1(100%)	8(100%)	2(100%)	1(100%)	100	Plate-shaped
Cross-in-tray	1(100%)	1(100%)	1(100%)	7(100%)	6(100%)	1(100%)	100	Multimodal, multiple global minima
Easom	1(100%)	1(100%)	1(100%)	⟨6⟩	3(100%)	1(100%)	100	Multimodal, multiple local minima, steep ridges/drops
Eggholder	2(100%)	1.3(92%)	1(100%)	⟨6⟩	1(100%)	1(100%)	100	Multimodal, multiple local minima
Goldstein-Price	1(100%)	1(100%)	2.4(100%)	13(100%)	1(100%)	1(100%)	100	Multimodal, multiple local minima
Holder Table	1(100%)	1(100%)	1(100%)	⟨6⟩	1(100%)	1(100%)	100	Multimodal
Levy-13	1(100%)	1(100%)	1(100%)	14(100%)	1(100%)	1(100%)	100	Multimodal
Matyas	1(100%)	1(100%)	1(100%)	7(100%)	1(100%)	1(100%)	100	Multimodal, one global minima, plate-shaped
McCormick	1(100%)	1(100%)	1(100%)	10(100%)	3(100%)	1(100%)	100	Multimodal, plate-shaped
Rastrigin	⟨5⟩	3.1(92%)	2.6(88%)	⟨4⟩	⟨6⟩	1(100%)	250	Multimodal
Rosenbrock	2.4(80%)	1.3(92%)	1.56(88%)	1.83(100%)	2.67(72%)	1(96%)	10400	Unimodal, valley-shaped
Schaffer-2	1(100%)	1.1(100%)	1.1(100%)	⟨6⟩	1.1(92%)	1(100%)	100	Multimodal
Schaffer-4	2(100%)	1.8(100%)	1(100%)	⟨6⟩	2.3(96%)	1(100%)	100	Multimodal
Sphere	3(100%)	1.2(100%)	1(100%)	58(100%)	1.4(100%)	1(100%)	100	Unimodal, bowl-shaped
Styblinski-Tang	1(100%)	1(100%)	1(100%)	⟨6⟩	1(100%)	1(100%)	300	Multimodal
CEC-2014 Unconstrained Benchmark Functions (Dimension=10)								
F1	60(100%)	1(100%)	1(100%)	8(100%)	⟨6⟩	1(100%)	1e+05	Unimodal, shifted, separable, scalable
F2	5(100%)	1(100%)	1.3(100%)	6.1(100%)	⟨6⟩	1.2(100%)	1.2e+05	Unimodal, shifted, non-separable, scalable
F3	117(100%)	1(100%)	1.4(96%)	65(100%)	⟨6⟩	1(100%)	1e+05	Unimodal, shifted, rotated, non-separable, scalable
F4	⟨5⟩	1.5(100%)	1(100%)	12.3(100%)	⟨6⟩	1.3(100%)	1e+05	Unimodal, shifted, non-separable, scalable, noise in fitness
F5	548.70(80%)	1(100%)	1(100%)	29.50(100%)	⟨6⟩	1(100%)	1e+05	Unimodal, non-separable, scalable
Avg. rank	2.65	1.6	1.7	5.26	4.13	1.1	-	Rank on all unconstrained functions
Constrained Functions								
Rosenbrock-1	⟨6⟩	1.3(88%)	1(100%)	⟨5⟩	1.36(80%)	1(100%)	2200	Two constraints, constrained with a cubic and a line
Rosenbrock-2	1(100%)	3.3(100%)	2(100%)	20(100%)	4(100%)	1(100%)	150	Single constraint, constrained to a disk
Simionescu	⟨6⟩	2.6(92%)	1.1(100%)	⟨5⟩	5(100%)	1(100%)	300	One constraint
Mishra	1(100%)	1.1(100%)	1(100%)	17(100%)	1.8(100%)	1(100%)	220	Single constraint
g11	2(100%)	1.75(100%)	1.2(100%)	3.1(100%)	4.6(90%)	1(100%)	200	Single equality constraint
g12	2.1(100%)	2(88%)	1(100%)	4(100%)	7.1(80%)	1(100%)	300	Eight inequality constraints
g13	10(100%)	1.66(100%)	1.1(100%)	7.2(100%)	12.3(70%)	1(100%)	600	Exponential function, three equality constraints
CEC-2017 Constrained Benchmark Functions (Dimension=10)								
C01	15.6(92%)	2.2(100%)	2(100%)	12.8(100%)	⟨6⟩	1(100%)	1e+05	Non-separable, single inequality constraint
C02	12.2(92%)	2.5(96%)	1(100%)	5.1(100%)	⟨6⟩	2.25(100%)	2e+05	Non-separable, rotated, single inequality constraint
C03	⟨6⟩	1.8(72%)	1.5(92%)	45(40%)	⟨5⟩	1(100%)	1.3e+05	Non-separable, one inequality and one equality constraints
C04	22.4(92%)	2(96%)	1(100%)	14.3(100%)	⟨6⟩	1.90(100%)	2e+05	Separable, two inequality constraints
C05	15.5(88%)	1.5(72%)	1.1(100%)	8.2(92%)	⟨6⟩	1(100%)	1.8e+05	Non-separable, two inequality constraints
Avg. rank	4.3	3.16	1.58	4.67	5.42	1.16	-	Rank on all constrained functions

For multimodal functions listed in Table 4.2, the results clearly indicate that MRFD can achieve convergence while optimizing these test functions with satisfactory accuracy. It can be observed that MRFD shows 1.1(minimum) to 58(maximum) folds improvement in solving multimodal functions with minimum number of FEs over traditional RFD and other peer algorithms. Figure 4.5 shows typical evolution curves of $Rastrigin$ function having four decision variables (dimension=4), which demonstrates a stable convergence behavior of MRFD over 25 runs, despite the occurrence of numerous local optima in this function.



(a) (b) (c)

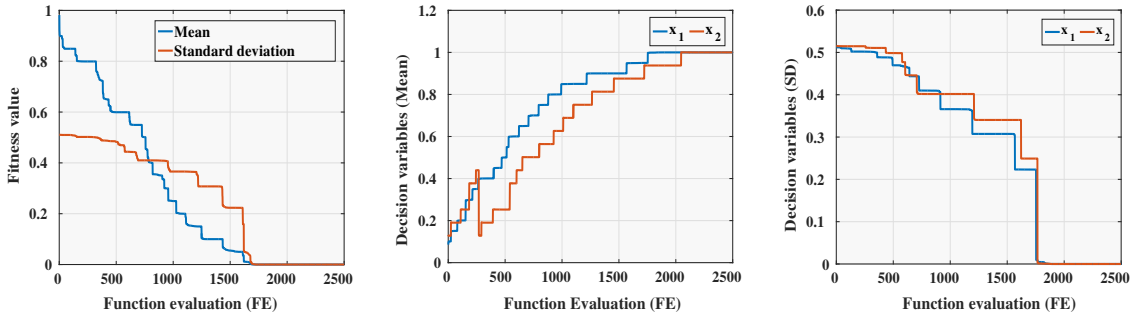
Figure 4.4: (a) Evolution of global fitness values, (b) mean and (c) standard deviation (SD) of ten variables of $F3$ test function at different function evaluations.



(a) (b) (c)

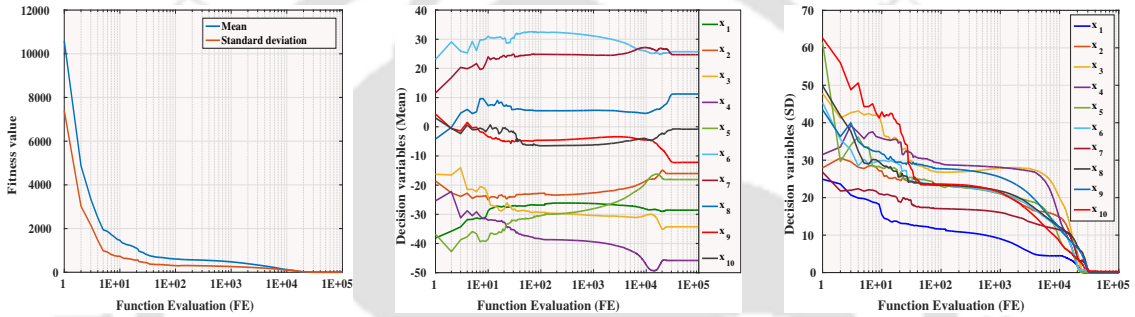
Figure 4.5: (a) Evolution of global fitness values, (b) mean and (c) standard deviation (SD) of four variables of $Rastrigin$ test function at different function evaluations.

Further, to demonstrate the applicability of proposed MRFD method on single objective constrained optimization problems, we consider 12 different constrained test functions ($Rosenbrock$-1, $Rosenbrock$-2, $Simionescu$, $Mishra$, $g11$, $g12$, $g13$ and $C01$-$C05$) as listed in Table 4.2. To handle any violations in constraints, Deb's rules described in [127] are followed to separate the occurrence of infeasible solutions from feasible ones. It can be observed that MRFD shows superior performance over traditional RFD and peer algorithms in solving all seven constrained test functions in terms of number of FEs. The proposed MRFD method solves $Mishra$, $Rosenbrock$-1, $Rosenbrock$-2, $Simionescu$, $g11$, $g12$ and $g13$ functions within 220, 2500, 150, 300, 200, 300 and 600 number of FEs, respectively. Moreover, it can be observed from Figure 4.6 that the evolution curves (mean and standard deviation) of fitness value and decision variables for $Rosenbrock$-1 function demonstrate a stable behav-



(a) (b) (c)

Figure 4.6: (a) Evolution of global fitness values, (b) mean and (c) standard deviation (SD) of four variables of $Rosenbrock$-1 test function at different function evaluations.



(a) (b) (c)

Figure 4.7: (a) Evolution of global fitness values, (b) mean and (c) standard deviation (SD) of ten variables of $C01$ test function at different function evaluations (FEs). FE is considered in log scale.

ior over several FEs, which showcase five fold improvement over traditional RFD method in terms of number of FEs. Moreover, for functions $C01$-$C05$, the results shown in Table 4.2 indicate that MRFD can solve these functions with higher success rates as compared to other peer algorithms. In addition, MRFD exhibits minimum FEs while solving these CEC-2017 constrained test functions as compared to other peer algorithms except for $C02$ and $C04$ test functions. Figure 4.7 shows typical evolution curves of $C01$ test function having ten decision variables (dimension=10), which demonstrates a convergence behavior of MRFD over 25 runs. It can be observed from Figure 4.7 that evolution curves of $C01$ test function achieve optimum values with minimum FEs (i.e., $\leq 1E + 05$).

As the evolution curves (mean and standard deviation) of fitness approach towards optimum value (i.e., zero), there appears to be a stable decrease in the fitness value with increase

in FEs. Such stable behavior of MRFD enables to achieve consistency in achieving global optimum solution. A composite ranking[8] is also evaluated to test the performance of all algorithms on test functions listed in Table 4.2 and it can be observed that MRFD has an average rank of 1.1 and 1.16 for unconstrained and constrained test functions, respectively, which are found to be the smallest among all. This demonstrates that MRFD maintains a consistent and satisfactory performance while solving the test functions.

The MRFD method can be extended in future to solve other complicated unimodal and multimodal functions [128–130] to demonstrate its applicability and effectiveness. However, as one of the motivations of this proposed work is to demonstrate the applicability of MRFD method in optimizing real-life single objective problems, we focus our discussion to minimize IR drops in VLSI power distribution networks.

4.4 Power Distribution Benchmarks

For demonstrating the applicability and effectiveness of proposed MRFD algorithm in minimization of IR drop issue, we have considered four IBM power distribution networks (i.e., $ibmpg2$, $ibmpg4$, $ibmpg5$ and $ibmpg6$) and six highly irregular industry standard power distribution networks (i.e., $industry1$-$industry6$[9]) as benchmark circuits. Complete descriptions of IBM power distribution benchmarks can be found in [113]. The benchmark circuits, $industry1$-$industry6$ are produced from internal designs of communication adapter (ring oscillator), memory controller and processors. The communication adapter ($industry1$ benchmark circuit) is created with a 2-level metal process, occupies a $22mm \times 22mm$ area, and uses an average wire width of $1\mu m$ at a pitch of $10\mu m$. Both memory controller and processor integrated circuits ($industry2$-$industry6$ benchmark circuits) are developed under $180nm$ CMOS process technology, which have 4 metal layers with different wire widths placed at a pitch of $9\mu m$ having more than 1 million nodes. All benchmarks are defined with DC loads.

[8]The ranks of all algorithms are evaluated depending upon the descending order of success rate and the ties are broken by ascending order of FEs.

[9]Provided by Semiconductor Lab, Chandigarh, INDIA.

However, for realistic model, the bond wires are also considered for industry benchmarks ($industry1$-$industry6$) during parasitic extraction, and are modeled as parasitic resistances. Table 4.3 lists details on all benchmark circuits considered for performance assessment.

Table 4.3: Power distribution benchmark statistics

Benchmarks	Benchmark Statistics[1]				IR drop profile (V)				Analysis (KLU Solver)
	$\#n$	$\#i$	$\#v$	$\#r$	Max.	Min.	Avg.	$\vartheta(\%)$[2]	Time (min:sec)
ibmpg2	127236	37926	330	208325	0.6160	0.00012	0.2860	47.36	00:06
ibmpg4	953581	276976	962	1560645	0.3860	0.00011	0.0051	36.18	00:10
ibmpg5	1079308	540800	539087	1076848	0.0520	0.00022	0.0127	35.89	00:21
ibmpg6	1670492	761484	836239	1649002	0.2170	0.00021	0.0227	42.06	00:35
industry1	250K	248974	1029	499000	1.6721	0.00020	0.1918	32.93	00:06
industry2	1M	998943	1059	1998000	1.0288	0.00020	0.5310	47.18	00:16
industry3	4M	3996980	3020	7996000	1.5315	0.00013	0.5721	53.71	02:02
industry4	9M	8994790	5211	17994000	1.6511	0.00021	0.5140	68.56	08.04
industry5	16M	15991481	8523	31992000	1.7013	0.00020	0.3856	76.57	22.39
industry6	25M	24987989	12013	49990000	1.7982	0.00018	0.2081	89.67	54.06

[1] $\#n$ denotes number of nodes, $\#i$ denotes number of current sources, $\#v$ denotes number of voltage sources, $\#r$ denotes number of resistances.
[2] ϑ denotes percentage of affected nodes (above threshold) before optimization.

4.5 Design Optimization Process

4.5.1 Preprocessing step

Before optimization, steady-state analysis is performed to generate the IR drop profile of the PDN benchmarks as presented in [131]. As the coefficient matrix ($\mathbf{A}_{r\mathcal{G}}\mathbf{G}\mathbf{A}_{r\mathcal{G}}^{T}$) representing the conductance network (steady state) is symmetric and positive definite [114], it can be simulated efficiently using existing power distribution network analyzers. During implementation, we employ KLU [132] for fast steady state analysis of power distribution network. The maximum, minimum and average IR drop values all PDN benchmarks are reported in Table 4.3 along with computational time involved in analyzing PDN benchmarks. It can be observed that with increase in number of nodes of power distribution networks, computational cost in analyzing these networks also increases, which affects the optimization process. Once analysis is performed, power distribution network is subjected to the optimization process for evaluating minimum IR drop and wire area subjected to various constraints.

4.5.2 Voltage drop at each node and minimization using modified river formation dynamics algorithm

Considering the power distribution network as a linear system as described in (4.1), the node potentials $\mathbf{v}_n$ are evaluated by performing either steady state analysis (on resistive network) or transient analysis (on RC/RLC-network) on the entire power distribution network before beginning the optimization process. This evaluation gives us a complete IR drop profile of the entire network as shown in Figure 1.4(a). For each node x as shown in Figure 4.1(c), the voltage drop corresponding to the metal widths of branches (e.g., $\rho l_{xy} g_{xy}$) or impulse current excitations (considered during transient analysis) at all the nodes is affected by the voltage drops at neighbor nodes corresponding to their connected metal widths and impulse current excitations. In fact, a node having IR drop affects its neighbor and, subsequently, affects a small region as shown in Figure 1.4(a). In view of this, the problem of IR drop at any node x can also be represented as (4.26),

$$\vartheta_x = V_{dd} - v_x = V_{dd} - \frac{\sum_{i=1}^{deg(x)} g_i v_i}{\sum_{i=1}^{deg(x)} g_i} + \frac{i_x}{\sum_{i=1}^{deg(x)} g_i}. \tag{4.26}$$

To minimize ϑ_x described in (4.26) subject to constraints in (4.5), $deg(x) + 1$ number of decision variables are considered which numerically constitute branch conductances (g_i, i=1 to $deg(x)$) and a current sink (i_x or I_{sink}). The proposed MRFD algorithm is designed to minimize IR drop of power distribution network as described in Algorithm 13. For each node, the neighbor branch conductances and current sink value are provided as a bound, i.e., being evaluated through steady state analysis (or transient analysis for RC/RLC power distribution network). The process of minimization starts by considering the decision variables as multiple volumes of sediment rates per unit time per unit width (q_s), which are varied in each time step using (4.13) guided by constriction coefficients (w_1, w_2, w_3) and probabilistic seed value (ε) as described in section 4.3.8. The constriction coefficients are generated within $(1, 2)$ as described in section 4.3.8) during evaluation. However, ε is evaluated for each decision variable by following the procedure described in Algorithm 12. Longitudinal slope (S_l) is

Algorithm 13: IR drop minimization procedure

input : A power grid network of size N.
output: Optimal IR drop at each node $x \in N$.
Data: $\mathbf{A}_{rG}\mathbf{G}\mathbf{A}_{rG}^{T}, \mathbf{A}_{rJ}\mathbf{i}_{J}$: Conductance and current sink matrices defined in section 4.2

```
while i < numExp do
    for each node x ∈ N do
        if V_dd − v_x < V_th then
            Minimize ϑ_x in (4.26) subject to constraints in (4.5)
            Evaluate optimal IR drop (ϑ_x) using MRFD
            /* update current node specification                    */
            Update optimal potential (v_x), branch conductances and current sink
            x ⟵ x + 1
        end
    end
    i ⟵ i + 1
end
```

evaluated from altitude values (h_1, h_2 as described in section 4.3.6) and distance between two altitude points ($x_1 - x_2$). During evaluation, the current potential at nodes represent different altitude and the conductances between neighbors correspond to distance. On the other hand, transverse slope (S_n) is evaluated using (4.11). Initial random generated values of decision variable (conductances or current sink) is assigned to q_s and, q_n and β are kept fixed at 1 and 1.4, respectively, for evaluation of S_n (positive slope). Further, u_s and u_n are assigned to potentials of current node and the neighboring nodes, respectively. In this manner, the slopes and corresponding probability values of all nodes are evaluated and considered during estimation of ε. Such implementation administers the evaluation of IR drop in optimal direction. Once the minimization is complete at a node, the corresponding node potential, current sink value and branch conductances are updated, and the process is shifted to another affected node (IR drop node) for minimization. This process of minimization is performed on the affected nodes for 25 times independently and the median of optimal IR drop is considered as the global optimum solution.

4.5.3 Minimizing metal area using MRFD method

Similar to the implementation of RFD scheme, MRFD method is employed to minimize metal area of power distribution network. The IR drop profile is evaluated before start of optimization process. Further, instead of analyzing a single branch of metal wire at time, several

cells are analyzed for the optimization of wire width. Wire widths (e.g., w_{xy}) are considered as decision variables (q_s), which are updated in each iteration by following (4.21). Similar to the minimization of IR drop, the values constriction coefficients (w_1, w_2, w_3) are set within $(1, 2)$ and the probabilistic seed value (ε) is evaluated using Algorithm 12. As similar constraints affect both IR drop and metal area, both random boundary handling strategy and constrained-dominance principle are employed for handling the constraints. A number of iterations are carried out on a single cell to evaluate the minimum width of each branch of the cell. Once the minimization is complete, corresponding cell is considered for the evaluation. The branches which are already being evaluated for minimum width (i.e., the overlapping branch between two cells) are not considered for further evaluation. The process of minimization at a single cell is performed for 25 times independently and the median of optimal width is considered as global optimum solution. In this way, wire widths corresponding to the affected nodes are optimized and total wire area of the power distribution network is evaluated at the end. As wire widths are evaluated in the optimal direction, after the evaluation, network area is also reduced in an optimal way. Such evaluation is important for designers as it serves as a guideline for power distribution network optimization.

4.6 Experimental Results

Both RFD and MRFD methods are employed to analyze each node of power distribution network for minimizing worst-case voltage drop (IR drop) and metal area. The process of minimization (of IR drop) is based on the framework of metal width and area (M_{area}) constraints (and IR drop constraints in case of minimization of metal area). Further, both the proposed methods can be extended to handle many other constraints, like transient current [133], timing yield [134] etc. without any additional computational overhead. Therefore, real-life (industry) power distribution networks are analyzed and optimized using both RFD and MRFD methods efficiently for minimum IR drop and reduced network area. Table 4.4 lists results of percentage nodes below threshold after minimization (ϑ') and percentage re-

Table 4.4: Comparison of proposed MRFD with SLP and RFD methods on different power distribution benchmarks

Benchmarks	SLP [32]				RFD [135]				MRFD			
	$\vartheta'(\%)$[1]	$A(\%)$[1]	Constr.[2]	Time (sec)	$\vartheta'(\%)$[1]	$A(\%)$[1]	Constr.[2]	Time (sec)	$\vartheta'(\%)$[1]	$A(\%)$[1]	Constr.[2]	Time (sec)
ibmpg2	0.91	03.90	72	98 (+ 0.23)	6.60	02.02	86	92 (+ 0.02)	1.06	04.00	89	39 (+ 0.02)
ibmpg4	1.84	06.00	162	913 (+ 0.50)	4.38	01.30	151	1146 (+ 0.11)	1.08	06.70	176	553 (+ 0.11)
ibmpg5	1.80	08.90	139	1221 (+ 0.62)	4.05	01.56	179	1427 (+ 0.30)	0.89	12.40	149	626 (+ 0.30)
ibmpg6	2.05	07.12	136	1734 (+ 0.67)	5.21	02.12	121	2201 (+ 0.31)	1.91	08.20	132	989 (+ 0.32)
industry1	1.24	03.95	200	258 (+ 0.45)	3.03	00.50	230	397 (+ 0.03)	0.06	06.56	312	189 (+ 0.03)
industry2	1.80	07.00	1123	1233 (+ 0.60)	4.03	02.02	1521	1310 (+ 0.32)	0.23	13.96	1035	571 (+ 0.32)
industry3	2.61	07.89	1406	3058 (+ 4.30)	6.13	03.34	2010	3732 (+ 1.23)	1.52	10.11	2256	1181 (+ 1.23)
industry4	-	-	-	-	9.25	04.56	2313	7310 (+ 5.23)	1.59	18.23	2751	1634 (+ 5.23)
industry5	-	-	-	-	12.40	06.23	3942	8897 (+ 12.34)	1.78	22.21	5127	2418 (+ 12.34)
industry6	-	-	-	-	-	-	-	-	1.98	28.63	5748	4394 (+ 23.41)

[1] ϑ' denotes percentage of affected nodes (above threshold) after optimization. 'A' denotes percentage of reduction in wire area after optimization.
[2] 'Constr.' denotes number of constraint violations during optimization. 'Time' denotes computational time (algorithm time + process time) required for entire optimization process.

duction in M_{area} (A) for different power distribution networks ($industry$ and $ibmpg$). The wire area (M_{area}) is evaluated using (4.4) by assigning a metal sheet resistance (ρ) of 0.01Ω and wire length (l) of $0.01\mu m$ for power distribution networks. A water drop population of 100 is selected to move the sediments (q_s) in search for optimum IR drop for a maximum function evaluations of 1000 for each node during implementation of MRFD method. Similarly, 100 water drop population is allowed to flow through altitudes to reach destination during implementation of RFD method. Two different types of benchmarks are analyzed and optimized using RFD and MRFD methods. It can be observed from Table 4.4 that MRFD outperforms RFD method both in minimizing IR drop and wire area reduction while optimizing $ibmpg$ and $industry$ benchmarks. Although $ibmpg$ benchmarks have highly irregular structures, number of affected nodes is reduced to 1.06%, 1.08%, 1.89% and 1.91%, and a reduction of 4.00%, 6.07%, 12.40% and 8.20% wire area is achieved for $ibmpg2$, $ibmpg4$, $ibmpg5$ and $ibmpg6$ benchmarks, respectively, after optimization using MRFD algorithm. Further, $industry1$-$industry6$ benchmarks are analyzed for optimization using MRFD algorithm. The percentage of number of affected nodes is reduced to 0.06%, 0.23%, 1.52%, 1.59%, 1.78% and 1.98%, and a reduction of 6.56%, 13.96%, 10.11%, 18.23%, 22.21% and 28.63% wire area is achieved for $industry1$-$industry6$ benchmarks, respectively, after optimization using MRFD algorithm.

Moreover, to demonstrate the applicability MRFD method, evolution curves (mean and

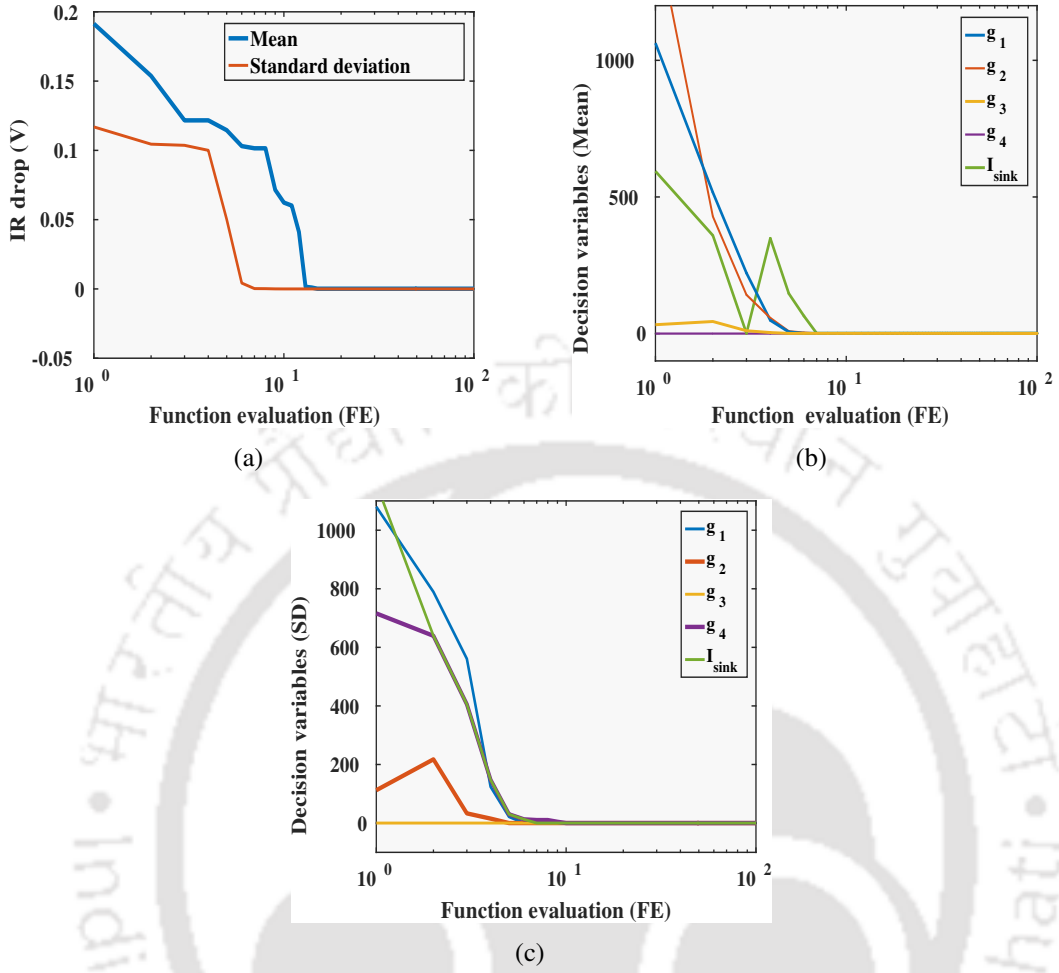


Figure 4.8: (a) Evolution of IR drop values, (b) mean and (c) standard deviation of five variables of a single node of $ibmpg6$ benchmark at different function evaluations (FEs). FE is considered in log scale.

standard deviation) of fitness value (IR drop) and decision variables (conductances and current sink) of a single node are shown in Figure 4.8 and Figure 4.9 for $ibmpg6$ and $industry6$ benchmarks, respectively. It can be observed from Table 4.4 that with increase in number of nodes in power distribution networks, the reduction in wire area and the affected nodes that exist after optimization using MRFD is random. This may be due to the probabilistic nature (dependence on probability during decision making) of MRFD method. Although the behavior of MRFD is random and is affected by probability, a minimum IR drop is achieved while considering metal area and current as constraints. To demonstrate the effectiveness of MRFD algorithm, a sequential linear programming (SLP) [32] is also implemented (author

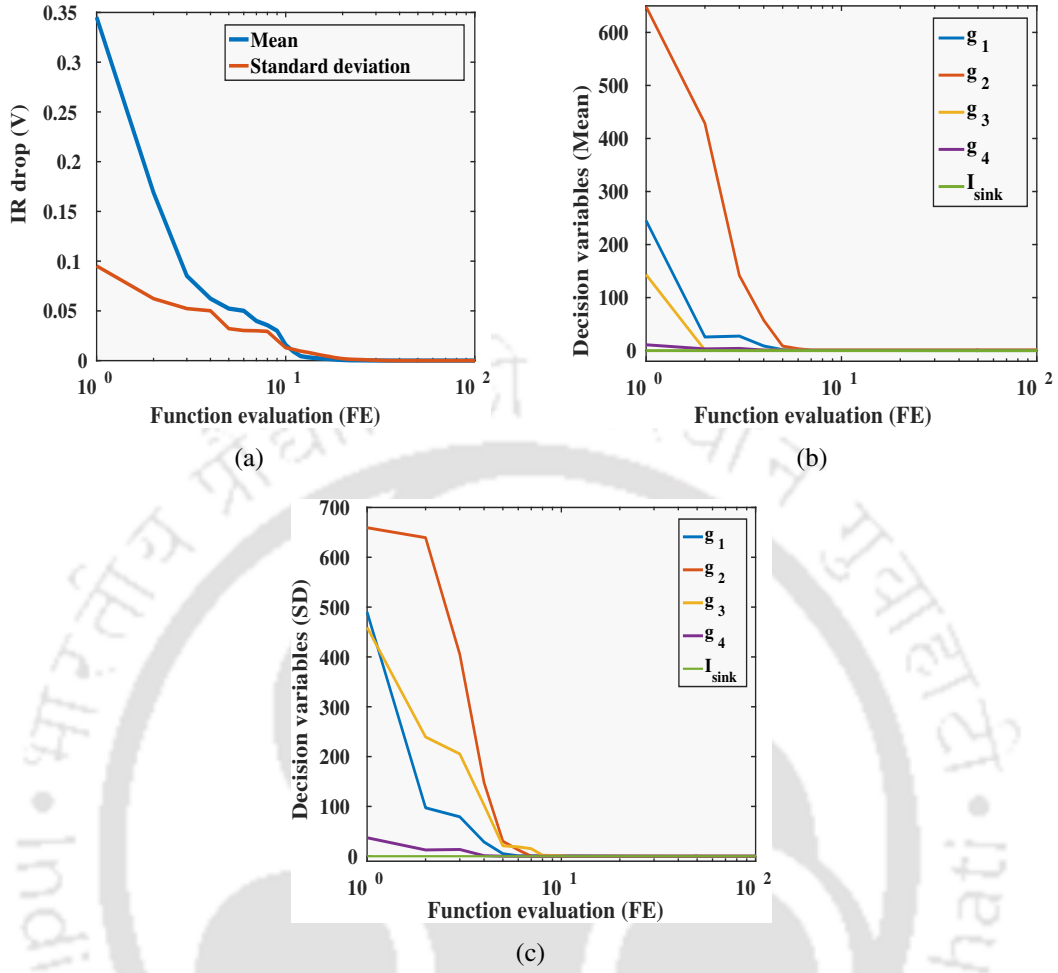


Figure 4.9: (a) Evolution of IR drop values, (b) mean and (c) standard deviation of five variables of a single node of $industry6$ benchmark at different function evaluations. FE is considered in log scale.

implementation of corresponding algorithms have been considered during evaluation) and is considered for optimization of similar power distribution benchmarks. It is to be noted that SLP has been used as a popular optimization tool in power distribution network design optimization. Table 4.4 lists number of affected nodes that exist and amount of reduction in wire area after optimization using RFD and SLP algorithms. It can be observed that SLP algorithm fails to achieve convergence for $industry4$ to $industry6$ benchmarks as the complexity and performance of SLP algorithm depend on the scale and network architecture [32]. Further, it can be observed that RFD also fails to converge for $industry6$ benchmark. As the optimization process is iterative, it requires rigorous execution of all methods to analyze all the nodes of power distribution network. Therefore, the computational cost in optimizing

the whole network is also evaluated and listed in Table 4.4 for all the algorithms. As RFD is implemented within similar framework of MRFD method, the process time (read time listed within brackets in Table 4.4) is similar for both the algorithms, while analyzing power distribution benchmarks. It can observed that MRFD method outperforms both RFD and SLP in computational cost while optimizing power distribution benchmarks under consideration.

4.7 Summary

In this chapter, two metaheuristics, i.e., RFD and MRFD are presented to minimize IR drop and wire area of power distribution networks. To improve the performance of RFD, two nature-inspired factors, such as sediment transport rate and transverse slope are introduced to RFD. The stability and convergence properties of MRFD method are analyzed to establish a convergence bound of constriction coefficients. The effectiveness of both RFD and MRFD are exhibited by the optimization of various standard single objective test functions and the results are compared with other peer metaheuristics. It is observed that MRFD outperforms these metaheuristics in terms of function evaluations and success rate. Further, the application of RFD and MRFD in the field of VLSI power distribution network is shown by minimizing IR drop and wire area of different industry standard benchmarks. Experimental results show that MRFD successfully minimizes IR drop and metal area. Based on the computation of proposed MRFD method, the knowledge of minimal IR drop and area reduction can be of significant advantage during early power distribution network design.

5

Analog/RF Circuit Design Optimization

Contents

5.1 Introduction

Analog circuit synthesis process is typically characterized by selecting a set of design parameters and topology. With the rapid progress in VLSI technology, manual design of an analog circuit is often confined to the usage of design automation for a fixed topology in finding a solution space, satisfying a set of design constraints. Due to high degree of nonlinearity and interdependence among design parameters, extensive computation is required to evaluate various design parameters during circuit design process. Although it is beneficial to reduce the computational cost associated with generating design performance estimates by modeling circuit design problem in the form of a linear programming problem and solving it using efficient optimization techniques, relevant emphasis is needed to analyze tradeoffs among various design specifications.

The advent of metaheuristics and their variants has been marked as a significant milestone towards evolution of automation in the field of analog circuit optimization [136], [137], [138]. Many metaheuristics are proposed in literature to efficiently analyze the analog circuit design problem [139], [140], [43], [141]. Metaheuristics inspired by analogies show stochastic behavior to some extent, which makes it possible to adhere to combinatorial explosion of possibilities in analog circuit design space. Many recently developed metaheuristics share a common problem of difficulty in adjustment of parameters according to suitable search space [138].

Therefore, in this chapter, a single objective optimization framework based on both RFD and MRFD methods (introduced in chapter 4) is presented in section 5.2 to pursue a path in a search space to find an optimal solution for analog/RF circuit design problem. Various experiments are performed on test circuits and the solutions are compared with other standard optimization techniques to showcase effectiveness of both RFD and MRFD methods. Further, because of complex circuit design process and the presence of uncertainties at many levels, analyzing a single cost function (single objective optimization) leads to faults and inconsistent solutions. This motivates the use of multiobjective optimization techniques during circuit

design to explore multi-dimensional design space (Pareto space) by correlating performances of multiple design specifications (Pareto fronts).

In view of this, we propose to analyze the tradeoff among performance specifications as a constrained-driven multiobjective optimization problem using three proposed multiobjective optimization algorithms in section 5.3. At first, a hierarchical version of NSGA-II (hNSGA-II) is presented in section 5.3.1 to perform analog circuit optimization using hierarchy in generation of new population during polynomial mutation operation. This section also presents a theoretical framework for hNSGA-II based on the asymptotic states of a homogeneous Markov chain model and proves the existence of stationary distribution with constant nonzero hierarchical mutation probability value (strictly positive). Further, a strong ergodicity bound is established to ensure the convergence to limit distribution asymptotically. For showcasing the effectiveness of proposed hierarchical scheme in polynomial mutation operation, six different mutation operators are incorporated in hNSGA-II framework and compared for convergence and diversity of Pareto optimal solutions. At second, a multiobjective optimization framework is developed using the proposed MRFD method, i.e., MOMRFD. The working principle of MOMRFD is discussed in section 5.3.2. At third, an improved version of MBSO algorithm is presented in section 5.3.3. The design examples and their respective analyses are discussed in section 5.4. Finally, the chapter is summarized in section 5.8.

5.2 Proposed Single Objective Optimization Algorithms for Analog/RF Circuit Optimization

5.2.1 River Formation Dynamics (RFD)

Since the basic structure of RFD scheme is already described in chapter 4, in this section, we present the implementation of RFD scheme for analog/RF circuit optimization. RFD starts by simulating water drops traversal from high altitudes to flat surface by transforming the landscape [84]. Water drops move to search for a destination by following a decreasing gradient principle. Each drop follows same or different paths from origin and the paths are

chosen probabilistically, which is proportional to the gradient of the down slope evaluated at each stage or node. Ever-decreasing slopes are quickly reinforced at a lower cost by sedimentation process, which assures free passage by avoiding any local cycles formed along the path. The complete RFD scheme [84] is described in Algorithm 7 (section 4.3.1). RFD scheme takes advantage of space locality to provide a way to reach destination. It accounts for all the possible combinations of level and proximal paths, and the possibility of flat surface. Therefore, it can be emphasized that the convergence of RFD scheme may be affected by the following possible cases.

(i) Drops follow flat surface infinitely many times.

(ii) Drops follow slopes infinitely many times. Following this case, finite combinations of paths may be generated to reach destination.

Proposition 1. *Let drops follow a flat surface infinitely many times, $f_0(x) \rightarrow f_\infty$ and every node is a solution to the problem, or the last visited node is a solution, if the number of nodes to be visited by drops is finite.*

Proof. If the first flat surface is encountered after k iterations, then $D_s(x_i, x_j) \rightarrow 0$, where x_i and x_j are two connected nodes, and the algorithm may stop at node x_i with an update $f_0(x) \leftarrow f_0^k(x_i)$. This update decreases the optimality gap $(f_0(x) - f_\infty)$ by a factor of $1 + \alpha_{x_i}^k$, where α is decided by probability of following the flat surface. If this occurs infinitely many times, then optimality gap will decrease further, which implies the assertion as $f_0(x) \rightarrow f_\infty$. To restart the search after encountering of the flat surface, a specific amount of $\delta_{i,j}$ is added to the altitude values of nodes. The value of $\delta_{i,j}$ is decided by the amount of erosion occurred during the path. This way, drops follow slopes to reach destination with finite combinations of paths. □

In RFD scheme, drops follow a path based on probability, which can be made arbitrarily close to one, if enough time is given, i.e., a guaranteed solution can be achieved. After a finite number of iterations, it is observed that the amount of erosion in connection to the optimal path is more with respect to other followed paths. This establishes the result that a potential optimal solution can be generated with finite combinations of paths.

5.2.1.1 Implementation details

Following a geographical metaphor, the flow of river to search for an optimum path to reach destination (sea) can suitably be applied to perform analog circuit optimization. In the proposed work, the search strategy of RFD is utilized to pursue the direction for finding an optimum solution of a single objective analog circuit design problem, i.e., circuit response. The movement of water drops from one node to another across a graph can be viewed as a movement of design variables (e.g., channel length, transconductance, output resistance etc.) towards convergence within the solution space. The overall optimization procedure is described in Algorithm 14.

Algorithm 14: Procedure for finding minimum cost function using RFD scheme

Input : Design parameter, x

Output : Minimum cost function, $\mathtt{F}_{\min}$

for $j \leftarrow 1$ to NUM_EXP **do**
  Generate random population of size N for design parameter x
  $total\_erosion \leftarrow \phi$
  $\mathtt{F} \leftarrow f(x)$ ▷ Evaluate population
  while *stopping criteria* **do**
    $r \leftarrow$ `random(0,1)`
    Find number of $x_{neighbor}$
    for $k \leftarrow 1$ to N_x **do**
      Select $x_{neighbor}$ at random ▷ N_x denotes number of neighbors of x
      if $r < Pr(x, x_{neighbor})$ **then**
        $break$
      end if
    end for
    $x_{next} \leftarrow x^k$ ▷ find the next
    $total\_erosion \leftarrow total\_erosion + Erosion(x, x_{next})$ ▷ update erosion using (**??**)
    if $x_{next} < x^L_{next} \,||\, x_{next} > x^U_{next}$ **then**
      $x_{next} \leftarrow x^L_{next} + rand(0,1) \times (x^U_{next} - x^L_{next})$
    end if
    $x_{next} \leftarrow x_{next} - total\_erosion$
    $x \leftarrow x_{next}$
    $\mathtt{F}' \leftarrow f(x)$
  end while
  if $\mathtt{F}' \not\preceq \mathtt{F}\ \forall\ x$ **then**
    $\mathtt{F}_{\min} \leftarrow \mathtt{F}$
  end if
end for

For initiating the process of optimization, a finite population is generated at random for

each design variable and the corresponding values are evaluated to determine the fitness function. The probable values of subsequent iterations are selected by evaluating probability at each point of selection for all the variables or design specifications, i.e., it is evaluated using (4.8), where altitude values correspond to the design specification or fitness function, which are to be minimized or maximized. The distance (weight) between two altitudes is evaluated by calculating square root of the Euclidean distance between corresponding design variables. Since the design variables change during iteration, corresponding fitness values are also evaluated in each iteration to determine slopes and probability. The altitude having maximum probability is selected as the next point in the path of the water drop and the fitness value is reduced by an amount of erosion using (4.9). The amount of erosion is considered as a penalty to the fitness function. The constraints are checked for any violation in each iteration using constrained-dominance principle [7].

Further, as excess erosion may lead to premature convergence and possible stagnation of design variables in the search space, the probable values of design variables are brought into the search space by using a boundary handling strategy as described in (5.1).

$$x \leftarrow x^L + rand(0,1) \times (x^U - x^L), \tag{5.1}$$

where x denotes the design variable; x^L and x^U represent the lower and upper bound of x, respectively, and $rand(0,1)$ evaluates a uniform random value between 0 and 1. The optimization process is continued until the design specifications meet certain limits or an iteration limit is reached. Finally, possible solutions are analyzed in each iteration and the solution having the minimum or the maximum value is selected as the best fitness value (i.e., $\mathtt{F}_{\min}$ as shown in Algorithm 14). In Algorithm 14, NUM_EXP is referred as the number of independent runs performed to carry out the optimization process. During execution of RFD, 25 independent runs are performed and the mean of best fitness values are chosen as the optimal solution.

5.2.2 Modified River Formation Dynamics (MRFD)

Although RFD method is capable of determining a near-optimal solution, there is a lack of efficient utilization of certain key factors in the original algorithmic model. Considering such cases, an improved version of RFD (MRFD) method is presented in the chapter 4. It is observed that MRFD has demonstrated better performance than other peer algorithms in solving standard single objective test functions and has been employed to minimize IR drop and area of power distribution networks successfully. As the details of MRFD is already given in chapter 4, only its application to analog circuit design optimization is discussed in this section.

5.2.2.1 Implementation details

Similar to the implementation of MRFD in section 4.5.2, the process of minimization or maximization of a fitness function starts by considering the design variables as multiple volumes of sediment rates per unit time per unit width (q_s). The q_s is updated in each iteration by using (4.13), which is guided by constriction coefficients (w_1, w_2, w_3) and probabilistic seed value (ε) as described in section 4.3.8.

The values of w_1, w_2, w_3 and ε are evaluated equivalently by following the procedure and various analyses described in chapter 4. During evaluation, S_l is evaluated from altitude values, which correspond to the current and previous fitness values, and the distance between two altitude points, which corresponds to the equivalent Euclidean distance between the current and previous values of design variables. Initially, randomly generated values of the design variables are assigned to different q_s values. Both q_n and β are kept fixed at 1 and 1.4, respectively, for the evaluation of S_n. Thus, the slopes S_l and S_n are evaluated including corresponding probability values to estimate the value of ε using Algorithm 12 as described in section 4.3.6. This implementation administers the evaluation of fitness function in each iteration and process is continued for a pre-defined number of generations for finding the effective solution (i.e., stopping criteria). Instead of selecting the final solution of a single experiment, 25 independent runs are performed and mean of best the fitness values is considered as opti-

Algorithm 15: Procedure for finding minimum cost function using RFD scheme

Input : Design parameter (q_s), constriction coefficients (w_1, w_2, w_3)

Output : Minimum cost function, $\mathtt{F}_{\min}$

for $j \leftarrow 1$ to NUM_EXP **do**
  Generate random population of size N for design parameter q_s
  $q_s^{rsed} \leftarrow \phi$
  for $i \leftarrow 1$ to N **do**
    if $f(q_s^{rsed}) > f(q_s^{lsed})$ **then**
      $q_s^{lsed} \leftarrow q_s^{rsed}$
    end if
  end for
  while *stopping criteria* **do**
    for $i \leftarrow 1$ to N **do**
      Generate random points between q_s^{lsed} and q_s^{rsed} to evaluate σ
      Evaluate ε using Algorithm **??**
      $q_s^{next} \leftarrow \varepsilon + w_1 \times (q_s^{rsed} - q_s) + w_2 \times (q_s^{lsed} - q_s) + \sigma \times w_3$
      $\mathtt{F} \leftarrow f(q_s^{next})$
      if $\mathtt{F} > f(q_s^{rsed})$ **then**
        $q_s^{rsed} \leftarrow q_s^{next}$
      end if
      if $f(q_s^{rsed}) > f(q_s^{lsed})$ **then**
        $q_s^{lsed} \leftarrow q_s^{rsed}$
      end if
    end for
  end while
  $\mathtt{F}_{\min} \leftarrow f(q_s^{lsed})$
end for

mal solution. The procedure of MRFD method for single objective analog/RF circuit design optimization is described in Algorithm 15.

5.2.3 Design Examples

5.2.3.1 Two-stage Operational Amplifier

In this section, a CMOS two-stage operational amplifier circuit is considered as a design example to showcase the applicability of both RFD and MRFD algorithms. The design problem is expressed as a single objective constrained optimization problem with an objective to maximize DC gain, A_v, of the two-stage operational amplifier circuit shown in Figure 5.1. The DC gain, A_v, of the amplifier can be represented as a function of design variables, such as transconductances ($g_{m1},\ g_{m2}, \ldots,\ g_{m7}$) and channel resistances ($r_{o1},\ r_{o2}, \ldots,\ r_{o7}$) of

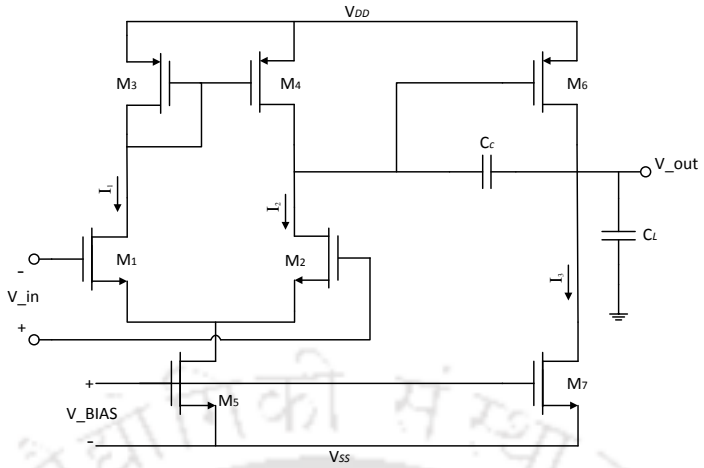


Figure 5.1: Two-stage operational amplifier circuit [142].

transistors in two-stage operational amplifier circuit as follows [142],

$$|A_v| \approx 2 \times \frac{g_{m2} \times g_{m6}}{(g_{ds2} + g_{ds4}) \times (g_{ds6} + g_{ds7})} \tag{5.2}$$

where, $g_{dsi} = 1/r_{oi}$, for $i = 2, 4, 6$ and 7. Details of the design constraints and specifications for two-stage operational amplifier circuit are listed in Table 5.1. The amplifier is designed in 0.5μm CMOS technology with 2.5V power supply using Mentor Graphics Pyxis tool [143]. The transconductances of different transistors are allowed to change from minimum to subsequent values within the limits of process technology. These transistors are matched according to appropriate relations ($M1 \equiv M2$, $M3 \equiv M4$) except $M5,\ M6$ and $M7$.

Table 5.1: Constraints and specification for two-stage operational amplifier

Constraints	Specification
Open-loop gain (A_v)	$Maximize$
Unity gain bandwidth (w_0)	$\leq 5\ MHz$
Slew rate (SR)	$>\ 10\ V/\mu s$
Input common-mode range ($ICMR$)	$-1\ to\ 2V$
Output voltage swing	$\pm\ 2V$
Load capacitance	$10\ pF$
Phase Margin	$\geq 60\,\deg$

For a single input, 25 independent runs are performed to demonstrate the quality and performance of both RFD and MRFD methods. For starting the process of optimization, two design variables (g_{m2} and g_{m6}) are considered. Their limits are imposed by evaluating all

design variables from values of constraints as shown in Table 5.1. Several experiments are performed on two-stage operational amplifier circuit to maximize the voltage gain subject to transconductance and output resistance of transistors, and the results are listed in Table 5.2.

Table 5.2: Performance Analysis of RFD for two-stage operational amplifier.

Methods	A_v^{best} (dB)	A_v^{avg} (dB)	A_v^{worst} (dB)	Iter. $(per\ run)$	Runs	Time (s) $(per\ run)$
PSO	79.18	79.08	65.6	10	15	0.03
SA	79.03	78.9	70.5	27	15	0.81
GA	78.69	78.13	74.86	12	15	0.04
DE	78.02	77.96	76.06	10	15	0.05
CRPSO	79.03	78.02	75.47	12	15	0.03
HPSO	79.60	79.04	73.45	17	15	0.75
RFD	77.81	77.23	76.10	19	15	0.3
ALC-PSO	79.81	78.01	72.05	15	15	0.5
MRFD	79.54	78.23	76.10	16	15	0.05
PYXIS	77.72	-	-	-	-	-

It can be observed from Table 5.2 that the occurrence of gain (best case) is around 77.81dB and 79.54dB while using RFD and MRFD methods, respectively. We have compared the solutions of RFD and MRFD methods with the solutions of other standard optimization techniques, e.g., particle swarm optimization (PSO) [67], simulated annealing (SA) [37], genetic algorithm (GA) [144], differential evolution (DE) [59], CRPSO [73, 74], HPSO [145] and ALC-PSO [78] to showcase the quality and performance of proposed RFD and MRFD methods. The results of these methods are also compared with the solution of Mentor Graphics Pyxis tool to verify the correctness of proposed methods. It can also be observed from Table 5.2 that the performances of both RFD and MRFD methods are competitive with the other standard optimization techniques in terms of achieving optimal solution for DC gain within reasonable amount of time.

5.2.3.2 Low Noise Amplifier (LNA)

Apart from the operational amplifier circuit, a low noise amplifier circuit (LNA) as shown in Figure 5.2 is also considered as a design example to demonstrate the effectiveness of both RFD and MRFD methods. For optimization of LNA design, noise figure (NF) is taken as an

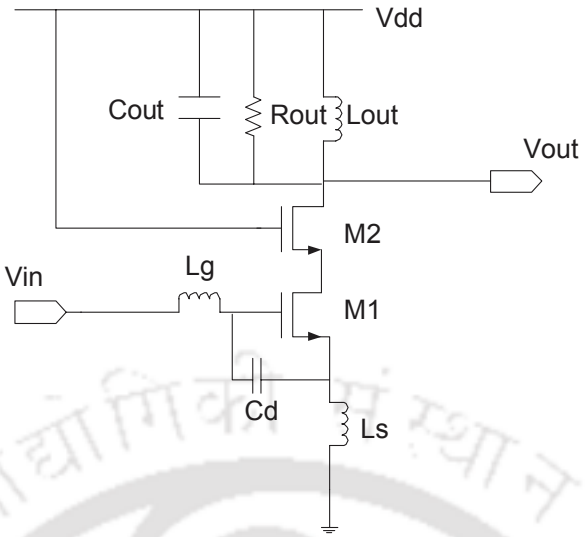


Figure 5.2: Low noise amplifier circuit [147].

objective function subjected to the constraints as shown in Table 5.3. The objective function, NF can be formulated as a function of width (W_{M1}), drain current (I_1) and capacitance (C_d). The absolute value of NF can be represented as [146],

$$NF = 1 + \frac{(A + B + C + D)}{E}, \tag{5.3}$$

where,

$$A = (1/4)\gamma g_{do},\ B = g_m^2(C_{gs}/C_{tot})^2(Q^2 + 1/4)\beta/5g_{do},$$
$$C = g_m c(C_{gs}/C_{tot})\sqrt{(\gamma.\beta)/20},\ D = 1/R_{out},\ E = g_m^2 R_s Q^2,$$

where γ denotes the white noise factor and is approximately equal to 1.05 for 180nm CMOS process technology; β represents the gate noise parameter and is approximately equal to 3.8; c expresses the correlation coefficient; C_{gs} denotes intrinsic gate capacitance; C_{tot} represents the sum of C_{gs}, C_d and parasitic capacitance,and Q denotes the quality factor of the input circuit. Approximated expressions of g_m and g_{do} are generated by curve-fitting [40], and they are expressed as follows,

$$g_m = A_0 L^{A_1} W^{A_2} I_{ds}^{A_3},$$
$$g_{do} = B_0 L^{B_1} W^{B_2} I_{ds}^{B_3}, \tag{5.4}$$

where A_0, A_1, A_2 and A_3 denote constants, whose values are equal to 0.0463, -0.4489, 0.5311 and 0.4689, respectively; B_0, B_1, B_2 and B_3 also represent constants, whose values are equal to 0.0096,-0.5595,0.5194 and 0.4806, respectively.

Table 5.3: Constraints and values for LNA design

Constraints	Values
Gain (Av)	$\geq 10dB$
Length of Transistor (M1,M2 and M3)	$L = L_{featuresize}$
Input Matching	$\frac{g_m L_s}{C_{tot}} = 50\ \Omega$
Current through M1 and M2(I_{ds})	$0.1mA \leq I_{ds} \leq 4.5mA$
Width of Transistors M1 (W_{M1})	$60um \leq W_{M1} \leq 100um$
External Capacitor (C_{ext})	$0.08pF \leq C_{ext} \leq 0.15pF$

Here, three different design variables, e.g., W_{M1}, I_1 and C_d, are considered for the evaluation of minimum NF. These design variables are bounded by limits as shown in Table 5.3. Several experiments are performed on LNA using peer algorithms to search for the minimum NF subject to width, drain current and drain capacitance. Since the objective of this optimization problem is to minimize NF, gain is maintained above 10dB to manage the tradeoff among NF and gain. The mean, best and worst case values of NF are listed in Table 5.4.

Table 5.4: Performance Analysis for low noise amplifier for a NF of 0.6 dB.

Methods	A_v^{best} (dB)	A_v^{avg} (dB)	A_v^{worst} (dB)	Iter. $(per\ run)$	Runs	Time (s) $(per\ run)$
PSO	13.80	12.03	10.62	20	15	0.10
SA	12.01	11.67	11.05	45	15	1.01
GA	13.62	12.11	10.86	27	15	0.17
DE	15.20	13.96	11.06	26	15	0.17
CRPSO	14.03	12.15	11.07	20	15	0.10
HPSO	14.60	14.02	13.45	40	15	0.92
RFD	14.81	14.23	13.60	19	15	0.60
ALC-PSO	14.72	14.01	12.05	30	15	0.30
MRFD	14.77	14.33	13.10	28	15	0.20
PYXIS	14.65	-	-	-	-	-

It can be observed from Table 5.4 that both RFD and MRFD methods have shown competitive performances within reasonable time bounds while minimizing NF of LNA, and minimum NFs (best cases) of 14.81dB and 14.77dB have been achieved using RFD and MRFD methods, respectively.

5.2.4 Discussion

One of the reasons for this competitive performance of RFD method may be due to the generalized tendency to select the paths by evaluating down slopes at each point during the movement of drops from origin to destination. This allows the drops to explore the structure of search space in parallel with the direction of movement. Further, due to the incorporation of additional factors in MRFD method, random movement of sediments enables the algorithm to explore promising regions of the search space in an aggressive manner. Consequently, suitable adaptation of constriction coefficients during movement and evaluation of random probability seed value favor the rate of convergence of MRFD method in a greater way.

5.3 Proposed Multiobjective Optimization Algorithms for Analog/RF Circuit Sizing

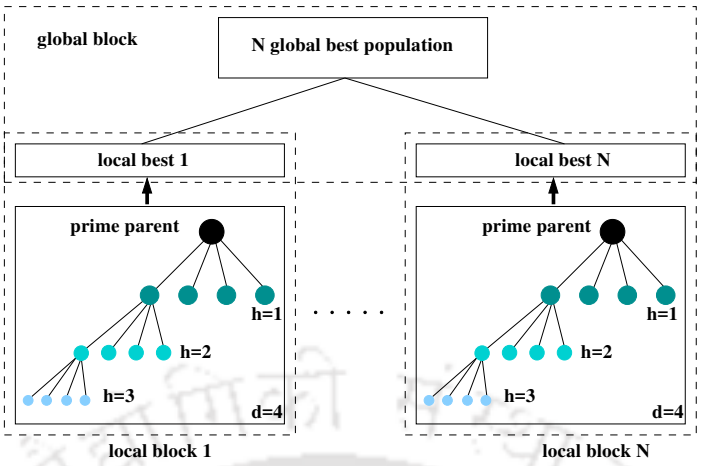
In the previous section, applicability of RFD and MRFD methods to solve single objective analog/RF circuit optimization is shown by maximizing DC gain A_v of a two stage operational amplifier circuit and minimizing NF of a low noise amplifier circuit. However, analog/RF circuit design problem comes up with a number of tradeoffs among design specifications or design constraints. In such cases, optimizing a single objective (cost function) of such design problems without considering the effect of other conflicting cost functions may not provide an efficient design solution of the circuit. Therefore, it is necessary to satisfy the behavior of contradictory objectives or cost functions by optimizing two or more conflicting cost functions simultaneously using suitable multiobjective optimization algorithms. Contrary to single objective optimization, the solution obtained by using multiobjective optimization algorithms is not confined to a single point solution, but a set of balanced solutions known as Pareto optimal solutions, which when plotted forms a Pareto front (PF) in the multiobjective search space [87]. The novelty of multiobjective optimization algorithm is to evaluate a set of solutions close to PF without the loss of convergence and diversity properties. These properties ensure to deal with nondominated solutions [7], while carrying out continuous design space

exploration. As mentioned in section 5.1, three new multiobjective algorithms are presented in this section to optimize two analog circuits and one RF circuit.

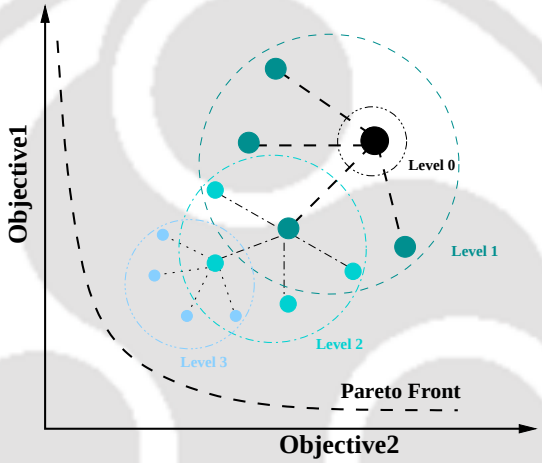
5.3.1 Hierarchical Nondominated Sorting Genetic Algorithm II (hNSGA-II)

A framework for hierarchical version of NSGA-II (hNSGA-II) is presented in this section. The proposed framework is based on the logic of conventional NSGA-II. Original NSGA-II makes use of a fast nondominated sorting approach with computational complexity $O(kN^2)$ [7], where k is the total objectives and N is the population size. Along with this elitist-preserving approach, NSGA-II uses crowding distance scheme to estimate the density of solutions for any specific PF. However, the formulation of this scheme becomes unstable when two or more solutions share a common fitness. Although NSGA-II keeps track of convergence by giving emphasis on nondominated solutions, there is scope of improvement in this category by employing suitable mutation or crossover strategies. The objective of our proposed approach is to improve the convergence property by incorporating hierarchy in the existing polynomial mutation scheme of NSGA-II. A generic hierarchical approach is presented for generation of child population during mutation operation, i.e., each child individual is formed in local neighborhood of its siblings along with parent population. The hierarchical approach for mutation operation is illustrated in Figure 5.3(a).

The child population is generated from corresponding parent population in a hierarchical structure, which forms local blocks of manageable size. Let us consider the generation of N local blocks and subsequent formation of one global block merging N local blocks and previous population. Each of the N local blocks is modeled as a tree like structure (having height h) with child population forming different branches. The first prime individual in a local block is called a *prime parent*. Prime parent generates a set of child individuals in the first level ($h = 1$) and the individuals in first level generate child individuals in the second level ($h = 2$) through mutation. Hence, the individuals in first level act as parents for individuals of second level (sub-parents) and so on. In this manner, child population in each local block is



(a)



(b)

Figure 5.3: (a) General model of hierarchical mutation strategy, (b) Hierarchical growth for $h = 3$ and $d = 4$ showing growth towards true PF.

generated using polynomial mutation operator and the process continues till maximum height of the tree ($h_{\max}$) is achieved. The number of child individuals of prime parent and each sub-parent is decided by the degree of branching d, and a total of β individuals are generated in each local block including the prime parent during each generation. The generated local best solutions in n^{th} local block having $d = 4$ moving towards PF in each level, is shown in Figure 5.3(b), where $n \in N$. Further, all child individuals in a local block along with prime parent are sorted and ranked using nondominated sorting approach. The individuals having highest ranks in each local block are selected as the local best individuals. These local best individuals are migrated to the global block to generate offspring population, which is merged

with parent population to generate a population having $2N$ individuals. These individuals are again sorted and ranked using nondominated sorting approach to select the best individuals in the next generation. However, to preserve the diversity during selection, it is necessary to select distinct local best individuals before migrating them to the global block without affecting the convergence property.

Instead of performing the ranking through traditional crowding distance approach, we propose to perform an indirect exclusive selection by performing hierarchical mutation operation iteratively. For an exclusive selection, at first N local best individuals are sorted in the descending order of crowding distance. All the local best individuals are checked for unique fitness and the individuals having unique fitness values (ε) are retained in the local best pool. The remaining $N - \varepsilon$ local best individuals are discarded and hierarchical mutation operation is carried out in corresponding $N - \varepsilon$ local blocks to generate new local best individuals. The process is continued till N local best individuals with distinct fitness values are generated. As the population in each local block is more than one ($\beta > 1$), where $\beta \leq \sum_i^h d^i$, the probability of generating local best individuals having distinct fitnesses is more as compare to traditional crowding distance method. In this regard, it accounts for the instability of crowding distance method in choosing common fitness values. While performing an indirect exclusive selection using hierarchical mutation operation, the number of individuals in each local block also play an important role in the convergence of solutions. A detailed theoretical framework of hNSGA-II along with convergence property is described in the following subsections.

5.3.1.1 Computational Complexity

As hNSGA-II is based on the framework of conventional NSGA-II, the algorithm starts by initializing parent population P_1 of the first generation. The hierarchical mutation procedure described in Algorithm 16, starts with initializing local block of height h and random generation of degree d for generating a temporary offspring population C_t. Once the number of offsprings in each level d is defined, the algorithm follows polynomial mutation operator to generate offspring population in each local block. The computational complexity of each

local block is $O(hd^j)$, where $j \in [1, h]$ number of computations in each block and a total of $O(hd^h(MN))$ number of computations (as j can go up to h) for generating the local offspring population C_t with M-dimensional objectives. The nondominated sorting in line 13 of Algorithm 16 having N population size requires $O(MN^2)$ computations [7].

Algorithm 16: HIERARCHICALMUTATION (P_t) procedure

```
1:  Initialize height (h) of each local block
2:  Initialize RAND_MAX (R_max) for random number generation
3:  d ← rand(R_max)                          ▷ R_max ∈ [1, 4]
4:  size ← N
5:  for i = 1 to size do
6:     j ← 1
7:     while j ≤ h do
8:        for l = 1 to d^j do
9:           C_t ← PolynomialMutation(P_t)          ▷ C_t → local block population
10:       end for
11:       j ← j + 1
12:    end while
13:    H_t ← NondominatedSort(C_t)
14:    cbest_t^i ← EliteChild(H_t)
15: end for
16: j ← 0
17: for i = 1 to N do
18:    for k = 1 to N do
19:       if |cbest_t^i| ≠ |cbest_t^k| then
20:          if |cbest_t^k| ∈ Q_t then
21:             continue
22:          else
23:             Q_t ← Q_t ∪ cbest_t^k
24:          end if
25:       else
26:          j ← 1
27:          while j ≤ h do
28:             for l = 1 to d^j do
29:                C_t ← PolynomialMutation(P_t)
30:             end for
31:             j ← j + 1
32:          end while
33:          H_t ← NondominatedSort(C_t)
34:          cbest_t^i ← EliteChild(H_t)
35:          go to step 19
36:          ε ← ε + 1
37:       end if
38:    end for
39:    ε ← 0
40: end for
```

As the search for unique fitness value among elite local individuals is recursive, it requires $O(\varepsilon(MN^2))$ computations with M-dimensional objectives. Once the local elite individuals are generated with the unique fitness values, they are migrated to the global block, which forms offspring population (Q_t) of t^{th} generation. This Q_t offspring population is combined with P_t to generate R_t having population size of $2N$. The nondominated sorting described in line 5 of Algorithm 17 having $2N$ population size requires $O(M(2N)^2)$ computations

[7]. Crowding distance assignment in line 9 requires $O(k(2N)log(2N))$ computations in the worst-case and quicksort involves $O(2Nlog(2N))$ computations, which speaks for an overall complexity of $O(M(\beta N + \varepsilon N^2))$ for hNSGA-II procedure, where $1 \leq \beta \leq (\sum_i^h d^i)$. With increase in the height of each block, runtime of hNSGA-II increases with an advantage of reaching convergence at minimum generations. One of the reasons for this functional advantage is the selection of an elite offspring from a pool of randomly generated individuals spanning alternative search spaces. Further, this ensures the evolution process to be promising as hierarchy in mutation operation facilitates the introduction of new offspring individual from random locations within the feasible space to preserve diversity in each generation.

Algorithm 17: hNSGA-II procedure

```
1:  Generate initial parent population P_1 of size N
2:  Q_1 ← HierarchicalMutation(P_1)            ▷ Q_t → offspring population
3:  for t = 1 to NGEN do
4:      R_t ← Q_t ∪ P_t
5:      F ← NondominatedSort(R_t)
6:      P_{t+1} ← φ
7:      j ← 1
8:      while |P_{t+1}| + |F_j| ≤ N do
9:          AssignCrowdingDistance(F_j)
10:         P_{t+1} ← P_{t+1} ∪ F_j
11:         j ← j + 1
12:         QuickSort(F_j)
13:     end while
14:     Q_{t+1} ← HierarchicalMutation(P_{t+1})
15: end for
```

5.3.1.2 Theoretical Framework

This section presents a theoretical framework for hNSGA-II based on asymptotic states of a homogeneous Markov chain algorithm. As hNSGA-II is based on NSGA-II, it uses the same basic operators such as selection, crossover and mutation to create next generation from current population. The generation of successive populations can be viewed as a stochastic process with finite state space, and each operation performed by the operators can be modeled as corresponding stochastic matrices. Further, the probabilistic changes of real parameters within a population using genetic operators can be formulated in the form of transition probability matrix P_{tr} [148], which can be represented as the product of stochastic matrices,

$$P_{tr} = P_s.P_m.P_c, \tag{5.5}$$

where P_s, P_m and P_c denote intermediate transitions caused by selection, mutation and crossover operation, respectively. Let us consider each genetic operation to be performed in succession. It is assumed that proportional selection is performed with probability P_s on an initial population vector $\bar{x} \in S$ having set of all possible population states S. The conditional probability $P_s(\bar{y}|\bar{x})$ of selecting a solution vector $\bar{y} \in S$ from initial population state vector $\bar{x} \in S$ can be represented as [148],

$$\begin{aligned} P_s(\bar{y}|\bar{x}) &= \frac{\Pi_{k\in S}\ f(y_k)}{(\Sigma_{k\in S}\ f(y_k))^n}, \quad \text{if } \bar{y} \subset \bar{x}, \quad (\bar{y}, \bar{x}) \in S \\ &= 0, \quad \text{otherwise} \end{aligned} \tag{5.6}$$

Mutation operation is performed to maintain the genetic diversity in successive generations by making changes to the states of decision parameters (variables) of current generation. As mutation operation is carried out independently on each individual decision variable by setting a user-defined mutation probability p_m to lower values (i.e. $p_m \in \{0, 1\}$), the conditional probability $P_m(\bar{y}|\bar{x})$ that the new population $\bar{y} \in S$ resembles the initial population $\bar{x} \in S$ after altering real values of variables at different decimal positions (mutation operation) can be aggregated to [148],

$$P_m(\bar{y}|\bar{x}) = p_m^{\text{poly}(x_k,y_k)}(1 - p_m)^{\text{poly}(x_k,y_k)}, \tag{5.7}$$

where $\text{poly}(x_k, y_k)$ is the polynomial mutation operator[1] [149] to generate child individual y_k from parent x_k. Typically, crossover operation is performed to generate child solution from two or more parent solutions. Here, simulated binary crossover (SBX) operator is used for crossover operation in hNSGA-II because of the classic property to keep difference between child individuals proportionate to their corresponding parent individuals. SBX coefficient provides the flexibility to control generation of child population in search space, i.e. child individuals having high SBX coefficients are more probable to provide solutions close to op-

[1]Polynomial mutation operation is considered as it is carried out using hierarchical scheme.

timum. Although SBX operator is used with crossover probability $P_c \in [0, 1]$, convergence analysis can be performed even without the knowledge of elements of transition crossover matrix. This is because of the analysis not to be affected by the choice of any specific operator.

The formal description of our proposed approach relies heavily on the hierarchical scheme of genetic evolution. The behavior of this scheme can be interpreted as a tree like homogeneous time bivariate Markov chain $\{(K_t, G_t), t \geq 0\}$, where K_t represents all individuals in the local block having height $h = 2$ and degree $d = 4$, and $G_t \in \mathbf{N}, G_t > 0$ as shown in Figure 5.4. Further, the prime parent is denoted by α and all other individuals in the local block are denoted by strings of integers in $[1, \beta]$, where $\beta \leq \sum_i^h d^i$. Moreover, behavior of the hierarchical mutation scheme implies few restrictions on the Markov chain for proper functionality. All transitions are considered to be of zero probability with the following two exceptions.

(i) At each step the process can make one transition to itself $(K_{t+1} = K_t)$ (no change in parent population after mutation operation) or,

(ii) Transition to one of its child $(K_{t+1} = K_{t+s}$, for some $1 \leq s \leq d)$ (change in parent population after mutation operation).

Local block of the hNSGA-II framework represents the evolution of child population from parent population using mutation operation. The transition probabilities of a parent to its children and to itself in one level may be different, even though both transitions are from a single individual. In view of this, Markov chain in each local block can be illustrated as the sum of probabilities of all possible sample paths in forward direction starting from prime individual and ending at child individuals of the final level. Therefore, the transition probability matrix of mutation can be rewritten as a combination of intermediate transitions described in (5.7) of each probable states. The transition probability matrix of mutation at

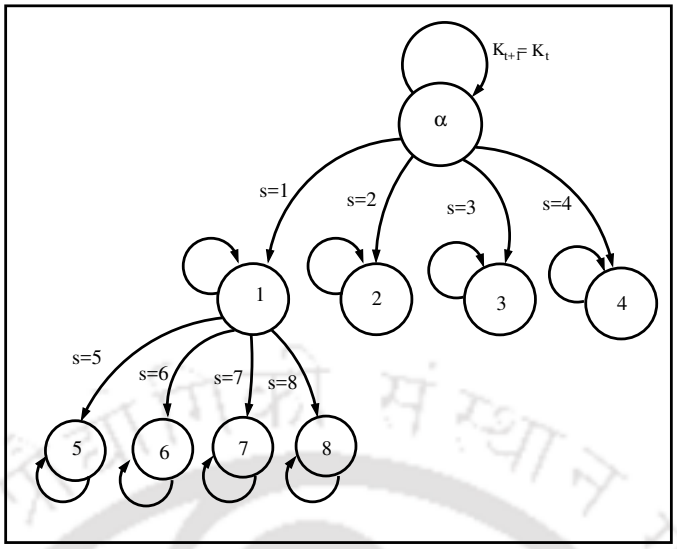


Figure 5.4: Markov chain model of hierarchical strategy

level h having degree d can be represented as,

$$P_m^{(h)}(\bar{y}|\bar{x}) = \Sigma_{l\in h}(\Sigma_{j\in d^l}\ p_m^{\mathrm{poly}_j^l(x_k,y_k)}(1-p_m)^{\mathrm{poly}_j^l(x_k,y_k)}) \tag{5.8}$$

It can be pointed out that the hierarchical mutation operator holds the key for robust convergence of hNSGA-II because it exhibits the decision whether the algorithm achieves limiting distribution (strong ergodicity) [150] or not. With the increase in the height h of each local block, the number of randomly generated individuals increases allowing the hierarchical mutation operator to explore additional search points in the feasible region. This augmented upper bound in mutation operation facilitates sufficient conditions for the strong ergodicity.

5.3.1.3 Convergence Analysis

Considering the search properties of hNSGA-II, various conclusions can be made relating to the sufficient conditions for hNSGA-II converging to a stationary distribution.

Proposition 2. *Any state probability distribution $\xi \in X$ of a homogeneous Markov chain is a stationary distribution in X if it satisfies the equality, $\xi^T = \xi^T P_{tr}$, where P_{tr} is the transition probability matrix and X is the state space containing all possible Markov states [148].*

Theorem 2. *For $p_m > 0$ and a strictly positive defined crossover and selection probability P_c, P_s, respectively, hNSGA-II converges in limit towards a stationary distribution ξ, where ξ*

is strictly positive, i.e. $\xi > 0$*, i.e.,*

$$\lim_{r\to\infty} P_{tr} = \begin{bmatrix} \xi \\ \xi \\ \vdots \\ \xi \end{bmatrix} \tag{5.9}$$

Proof. Let (x_t, x_{t+1}) be the given arbitrary states at any time point $(t, t+1)$ respectively before hierarchical mutation. The transition from x_t to x_{t+1} in one time step with appropriate probability P_m can be given by (5.8). Let us assume the corresponding relative fitness be $f_{(t,t+1)}$. Depending upon the number of states in each block, there are two possible cases.

Case 1: For $x_{t+1} = 0$, there exists exactly one state with a relative fitness $f_{(x_t,x_t)} > 0$ for population of fixed size N as the state transits to itself every time. It can be concluded that,

$$P_m^h(\bar{x_t}|\bar{x_t}) = p_m^{\text{poly}_j^l(x_t,x_t)}(1-p_m)^{\text{poly}_j^l(x_t,x_t)} > 0 \quad (l,j=0) \tag{5.10}$$

Case 2 : For $x_{t+1} > 0$,

$$P_m^h(\bar{x_{t+1}}|\bar{x_t}) = \Sigma_{l\in h}(\Sigma_{j\in d^l}\ p_m^{\text{poly}_j^l(x_{t+1},x_t)}(1-p_m)^{\text{poly}_j^l(x_{t+1},x_t)}) > 0 \quad (l,j>0) \tag{5.11}$$

□

Therefore, for arbitrary states $(x_t, x_{t+1}) \in X$, there exists strictly positive numbers in P_m^h. As both crossover and selection transitions are strictly positive, the entry in P_{tr} consists of a product of strictly positive numbers. □

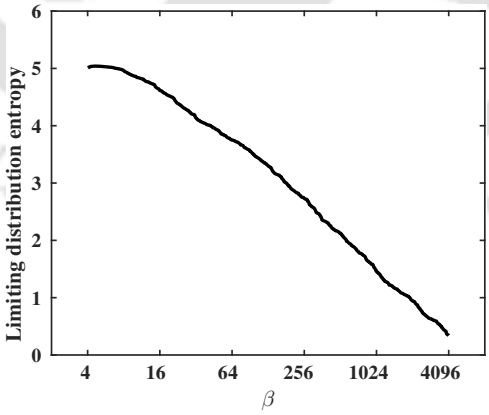


Figure 5.5: Variation of limiting distribution entropy with local block population β.

As the stationary distribution is strictly positive, all states are considered to be associated with finite probability. This means that a limiting behavior may be possible and to showcase

the possibility, it is important to evaluate the entropy with respect to the increase in population size inside each local block during mutation operation. The usefulness of hierarchical mutation scheme is shown in Figure 5.5 by presenting variations in the limiting distribution entropy [148] with respect to β. The results illustrate the approximate computed stationary distribution entropy on a test problem (ZDT1) at different selected values of β. The values of β are evaluated by considering $d = 4$ and $1 \leq h \leq 6$. It can be observed from Figure 5.5 that the entropy reduces to zero with increase in β. Further, it can be suggested that with increase in β, it may be possible to approach limiting distribution behavior ($i.e.$, a globally optimal solution with probability one) by making the probability associated with nonoptimal states as small as required. Mutation operation is performed for both polynomial and binary coded genetic evolution in such a way, that it enforces the relative fitness ($f > 0$) [151], which may parameterize hNSGA-II to reduce bias towards solutions by analyzing ergodic bound of hierarchical mutation probability. The presence of uncertainty in transitions during hierarchical mutation to choose between next state and the current state may cause finite number of revisits to already traversed state exhibiting transient states ($P_m^{(h)}(\bar{y}|\bar{x}) < \infty$)(Theorem 3). Further, it may account for few states to be recurrent ($P_m^{(h)}(\bar{y}|\bar{x}) = \infty$), if the state transits to itself in all transitions. In such cases, variations in population size parameter β inside local block of hNSGA-II has important implications for recurrence and transience conditions.

Proposition 3. *In any state space X, a state x_t is recurrent if and only if $\sum_m^\infty P_m^h(\bar{x}_t|\bar{x}_t) = \infty$, transient otherwise. [151]*

Theorem 3. *All states in X of a Markov chain are either recurrent or all states in X are transient. Thus, if there is a transition from x_t to x_{t+1} and state x_t is assumed to be recurrent, then x_{t+1} is also recurrent. Equivalently if a transition occurs from state x_t to x_{t+1} and state x_t is transient, then state x_{t+1} is also transient. In particular, it can be said that for an irreducible Markov chain, either all states are recurrent or transient. [151]*

Proof. If there is a transition between x_t and x_{t+1}, and x_t is recurrent, then x_{t+1} must be recurrent because every time x_t is revisited after some interval with a success probability, x_{t+1} also gets revisited with a finite time interval. Since x_t is recurrent, it may be visited a number of times with a probable success rate making x_{t+1} to be recurrent. Eventually, this can be viewed to be a sequence of random walk repeating itself at regular intervals. □

5.3.1.4 Conditions for Ergodicity

Mutation operation plays a prominent role in robust convergence of a simple genetic algorithm. With inclusion of hierarchical mutation operation, the number of individuals participating in genetic evolution process is comparatively more than NSGA-II, which in turn increases the search space of hNSGA-II and ensures a rigid schedule bound for ergodicity. In this section, we present a schedule bound for hierarchical mutation in hNSGA-II that establishes strong ergodicity. There are a number of conditions to showcase the existence of ergodicity in the proposed Markov chain. One of the necessary and sufficient conditions for ergodicity is the existence of a non-trivial measure [152], μ satisfying (5.12).

$$\mu_m^h(\bar{y}) \geq \int_X \mu_m^h(d\bar{x}) P_m^h(\bar{y}|\bar{x}) \tag{5.12}$$

According to the general state space context, it is quite difficult to verify the criterion presented in (5.12). Therefore, a Doeblin condition [153] can be established along with the condition in (5.12) satisfying (5.13) having a probability measure (μ) and a fixed integer δ, s.t. $0 < \delta \leq \beta$, that may be accounted for a schedule bound for ergodicity of the proposed Markov chain model, i.e.,

$$P_m^h(\bar{y}|\bar{x}) \leq \delta, \forall \bar{x} \in X \tag{5.13}$$

Theorem 4. *Following both conditions in (5.12) and (5.13), a sufficient condition for the proposed Markov chain to be ergodic is the existence of a non-trivial measurable function g such that [152],*

$$\int_X P_m^h(d\bar{y}|\bar{x}) g_m^h(\bar{y}) \leq g_m^h(\bar{x}) - 1, \forall \bar{x} \notin X \tag{5.14}$$

and for a fixed integer $\delta > 0$,

$$\int_X P_m^h(d\bar{y}|\bar{x}) g_m^h(\bar{y}) = \omega(\bar{x}) \leq \delta < \infty, \forall \bar{x} \in X \tag{5.15}$$

Proof. Considering g to be a non-measurable function as described in theorem 4 and let there is a transition from state $\bar{x}$ to state $\bar{y}$ using mutation probability, P_m^h having height h of a each local block, then at $(t+1)^{th}$ time point,

$$g_{m(t+1)}^h(\bar{x}) = \int_X P_{m(t)}^h(d\bar{y}|\bar{x}) g_m^h(\bar{y}), t \geq 1$$

If there exists an intermediate state space $\kappa, \kappa^c \in X$, such that $X = \kappa \cup \kappa^c$ then

$$\begin{aligned} g^h_{m(t+2)}(\bar{x}) &= \int_X P^h_{m(t)}(d\bar{z}|\bar{x}) \left[\int_X P^h_{m(t)}(d\bar{y}|\bar{z}) g^h_m(\bar{y}) \right] \\ &\leq \int_\kappa P^h_{m(t)}(d\bar{z}|\bar{x})\omega(\bar{z}) + \int_{\kappa^c} P^h_{m(t)}(d\bar{z}|\bar{x})[g^h_m(\bar{z}) - 1] \\ &\leq \int_\kappa P^h_{m(t)}(d\bar{z}|\bar{x})[\omega(\bar{z}) + 1] + \int_{\kappa^c} P^h_{m(t)}(d\bar{z}|\bar{x}) g^h_m(\bar{z}) - P^h_{m(t)}(X|\bar{x}) \\ &\leq (\delta + 1) P^h_{m(t)}(\kappa|\bar{x}) + g^h_{m(t+1)}(\bar{x}) - 1 \end{aligned} \tag{5.16}$$

Iterating (5.16) and dividing by t, we have

$$\begin{aligned} \frac{1}{t} g^h_{m(t+2)}(\bar{x}) &\leq (\delta + 1) \left[\frac{1}{t} \Sigma^t_{i=1} P^h_{m(i)}(\kappa|\bar{x}) \right] + \frac{1}{t} g^h_{m(2)}(\bar{x}) - \frac{1}{t} \Sigma^t_{i=1} 1 \\ &\leq (\delta + 1) \left[\frac{1}{t} \Sigma^t_{i=1} P^h_{m(i)}(\kappa|\bar{x}) \right] + \frac{1}{t} g^h_{m(2)}(\bar{x}) - 1 \end{aligned} \tag{5.17}$$

As per [152], for $\forall x$,

$$\lim_{t \to \infty} \frac{1}{t} \Sigma^t_{i=1} P^h_{m(i)}(\kappa|\bar{x}) = \pi(\kappa|\bar{x}) \tag{5.18}$$

exists and $\pi(\kappa|\bar{x})$ is finite when $\kappa \in X$. With increase in time point as $t \to \infty$ and by the non-negativity of $g^h_{m(t)}$ and finiteness of $g^h_{m(2)}$ in (5.17), we have

$$\pi(\kappa|\bar{x}) \geq \frac{1}{\delta + 1} \geq \frac{1}{\beta} > 0 \tag{5.19}$$

where for a finite population size β, there exists a schedule bound for ergodicity of the proposed Markov chain model. □

5.3.1.5 Performance Analysis

- **Parametric study :** In this subsection, we perform a parametric study for population size parameter β of hierarchical polynomial mutation operator. Since the performance of proposed hierarchical scheme can be characterized for different population sizes of each local block (β), the performance can change with the change in β. For a constant degree d, height h can be increased to postulate variations in both convergence and diversity of Pareto optimal solutions at different values of β for a number of generations. To measure the performance in terms of convergence towards optimal Pareto front and

diversity of solutions along Pareto front, three different quality metrics[2] are evaluated, i.e. *Generational Distance* (GD), *Inverted Generational Distance* (IGD) and *Hypervolume* (HV). GD metric [98] is used to measure minimum Euclidean distance between generated final individuals and sample points on true Pareto front. Smaller value of GD indicates a better convergence towards true Pareto front. On the other hand, both IGD metric [98] and HV metric [154] are a measure of both convergence and diversity of solutions along true Pareto front on a single scale. If the IGD value is close to zero, it means that the solutions have better convergence and diversity, and HV is a fair measure of the maximum area covered by the nondominated solutions with respect to a reference point.

In view of this, several experiments are carried out on ZDT1 test function for a number of generations and it can be observed from Figure 5.6 that with increase in β, solution (optimal PF points) attempts to attain smaller (best) GD (Figure 5.15(a)) and IGD (Figure 5.15(b)) values and higher HV (Figure 5.15(c)) values at fewer generations. One of the reasons for these best values at fewer generations for higher β value can be due to the increase in number of offsprings in each local block with increase in β. It can be seen from Figure 5.15(c) that with the increase in number of generations, HV values also increase. However, the HV values remain unchanged after 500 number of generations (for $\beta = 256$ and 700 generations for other β values). Since intermediate traces of PF exist at fewer generations, the population has to be evolved over sufficient number of generations (500 generations for $\beta = 256$) to account for the convergence and diversity metric to saturate during function evaluations. With more population, the diversity can be preserved by exploring extra search space by combining all new offsprings of each local block. Another reason for the best values at fewer generations can be the reduction in selection pressure by generating more offsprings through proposed hierarchical mutation strategy. With decrease in selection pressure, the convergence

[2] The details of quality metrics are described in Appendix C

issues are addressed to a great extent by mitigating the instability in crowding distance method. However, as we know, β cannot be increased to any arbitrary value. Since β accounts for the variations in limiting distribution entropy of mutation operation, it should be kept within finite limits ($0 < \beta < N$) for better performance of hNSGA-II (as described in section 5.3.1.4).

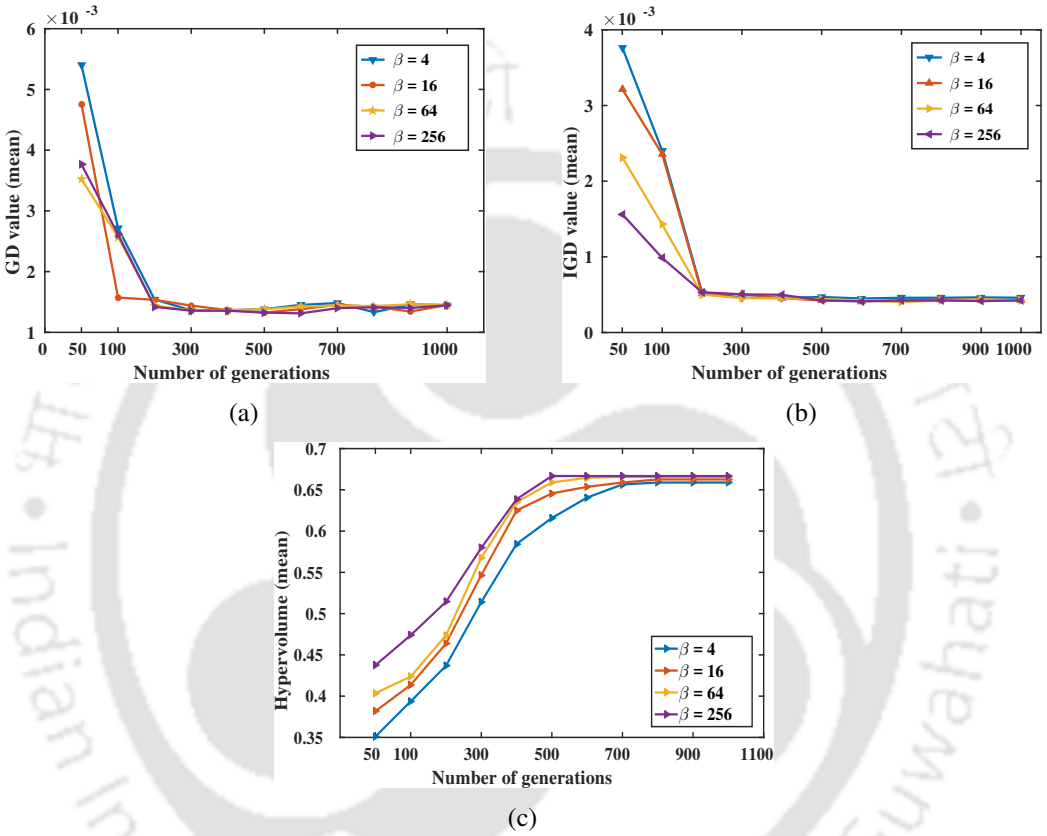


Figure 5.6: Performance of hNSGA-II on ZDT1 test function at different generations for various population sizes of local block (β), (a) mean GD values (b) mean IGD values (c) mean HV values.

- **Comparison with Other Mutation Strategies :** As mutation operator operates on an individual independent of other population members, it plays an important role in making overall search efficient [155]. Therefore, it is necessary to investigate the effectiveness of proposed hierarchical mutation operator by comparing the performance with other mutation operators. Hierarchical scheme for six different mutation operators (Adaptive Levy mutation [156], Cauchy mutation [157, 158], uniform mutation [159], nonuniform mutation [159], Gaussian mutation [160] and Adaptive-mean

mutation [157]) are implemented and also incorporated to study the search efficiency of hNSGA-II framework. This study is extensive in evaluating the performance of proposed hierarchical polynomial mutation operator. Further, a total 30 decision variables ($p = 30$) are considered during evaluation and each variable is bounded within $x_i \in [0, 1]$, where $1 \leq i \leq 30$. Crossover probability is kept at 0.9 and mutation probability is set to $1/p$ for all mutation schemes. Both crossover and mutation distribution indices are set to 20 during the entire process. Since various mutation operators are employed within hNSGA-II framework, different parameters are addressed each time a mutation operator is exercised. During Cauchy mutation, a Cauchy density function [158], $f_{cauchy} = \frac{1}{\pi}\frac{t}{t^2+x^2}$ is employed, where scale parameter t is set to 1 during operation. The parameter α fixed at 1.7 for all experiments while using adaptive Levy mutation scheme. Box-Muller transform is adopted to generate Gaussian distribution from uniformly distributed numbers during Gaussian mutation operation. Population size of each local block, β is kept at 256 for achieving hierarchy during all operations.

All mutation operators exercised using hierarchical scheme are compared based on convergence and diversity of solutions (mean GD values, mean IGD values and mean HV values), while 200 population individuals are allowed to evolve over 100 generations on ZDT1 test function. For each mutation operation, 30 independent runs are performed and the results are shown in Figure 5.7. It can be observed from Figure 5.7(a) and Figure 5.7(b) that proposed hierarchical polynomial mutation operator performs better than other mutation operators as corresponding GD and IGD values are minimum. Further, it is observed from Figure 5.7(c) that the proposed hierarchical polynomial mutation operator has larger HV value as compared to other mutation operators. As the proposed hierarchical scheme is performed on top of polynomial mutation operator, the inherent probability distribution of creating an offspring population is similar to polynomial mutation operator, which is instrumental in the convergence of Pareto optimal solutions. Moreover, the implicit generation of distinct fitness values in each generation favors the

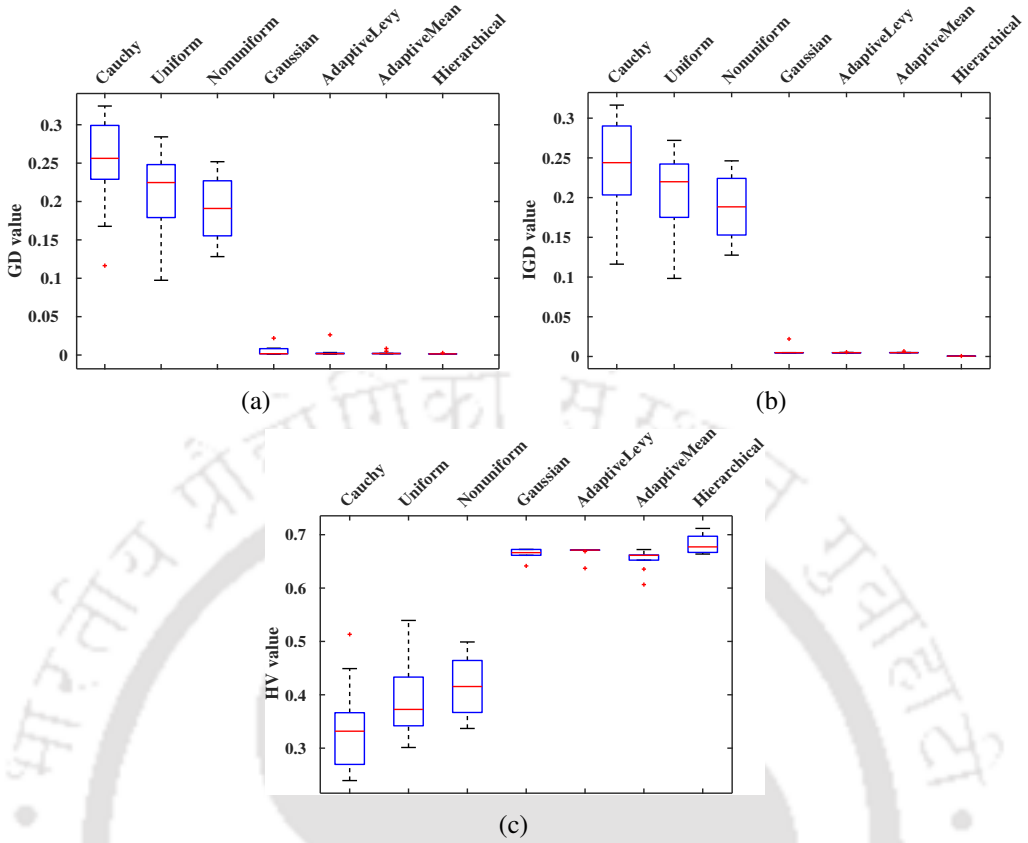


Figure 5.7: Performance analysis of several mutation strategies on ZDT1 test function showcasing mean, (a) GD values (b) IGD values (c) HV values at different generations (100 to 1000).

preservation of diversity among Pareto optimal solutions.

5.3.2 Multiobjective Modified River Formation Dynamics (MOMRFD)

In this section, a multiobjective framework based on MRFD method, i.e., MOMRFD is presented. Similar to single objective framework, MOMRFD is also a population based multiobjective optimization technique, where each population individual corresponds to sediment deposition rate per unit time per unit width (q_s). The main two attributes of MOMRFD method are given below.

- Altitude values denote the current position of water drops in a geographical terrain.
- Decision variables of an optimization problem correspond to sediment deposition rate per unit time per unit width, q_s.

Algorithm 18: Procedure for multiobjective modified river formation dynamics (MOMRFD)

Input : A multiobjective optimization problem (MOP), A stopping criteria, Number of population individuals (N), Archive size (K), maximum number of generations (MAX_GEN).
Output : The nondominated solutions, N available in external `archive`, objectives $\mathbf{F}_{\min}$.

1: Generate an initial random population, q_s^i, where $i \in N$;
2: **while** $g < MAX\_GEN$ **do**
3: Evaluate constraints, i.e., both boundary constraints using random constraint evaluation method [115] and constraint functions using constraint dominance principle [7];
4: Evaluate fitness for each individual q_s, $\mathbf{F} = f(q_s^i)$, $i \in N$. and update `archive` according to Pareto dominance;
5: Generate new population individuals, q_s^j from q_s^i, where $i, j \in N$ using modified RFD method as described in 4.21, i.e., employ modified probability and new sedimentation parameter, ε during analysis;
6: Evaluate constraints as described in step 3.
7: Evaluate the new population, $\mathbf{F} = f(q_s^j)$ and apply Pareto dominance to generate nondominated solutions, $\mathbf{F}_{\min}$;
8: **if** $\texttt{archive}_{size} < K$ **then**
9: Add nondominated solutions to `archive`;
10: **end if**
11: **if** $\texttt{archive}_{size} \geq K$ **then**
12: Sort nondominated solutions and rank them using crowding distance scheme [7], i.e., `archive` is maintained by keeping the individuals having better nondominated solutions using crowding distance scheme followed by niche count [7];
13: Employ polynomial mutation operation [7] to generate new individuals from best individuals (i.e., individuals having best fitness value (q_s^{lsed})) in `archive` and replace the best individual if the mutated individual has better fitness value;
14: **end if**
15: **if** stopping criteria satisfied **then**
16: The nondominated solutions in `archive` are $\mathbf{F}_{\min}$.
17: break;
18: **else**
19: goto step 5 ;
20: **end if**
21: $g \leftarrow g + 1$
22: **end while**

The MOMRFD method begins the optimization process by initializing the altitudes corresponding to each drop in the population with an equal non-zero random value. In the first step, a definite population (N) of decision variables (q_s) are generated at random. Each individual in the population is evaluated to obtain a solution to the cost function and the solution having the best cost function value is regarded as regional best (q_s^{lsed}) solution. All initial solutions are treated as nondominated solutions and are stored in an `archive`. The decision variables (q_s) are updated in each iteration using (4.21) and the corresponding solutions are evaluated as described in Algorithm 18.

The process is continued for a number of iterations, which is referred as generations. The

selection of global best solution in each iteration is imperative to achieve convergence and diversity of nondominated solutions. Since a multiobjective optimization problem has a set of optimal solutions, called Pareto optimal solutions, the solutions obtained in each iteration are added to `archive`.

As mentioned above, MOMRFD makes use of an external `archive` to store nondominated solutions. The criteria based on which the `archive` is maintained, plays a pivotal role in ensuring both convergence to true pareto-optimal front as well as diversity of solutions. As new nondominated solutions are generated in each generation, they are appended to the `archive`. When the `archive` is full, it is necessary to maintain the `archive` size by selecting superior nondominated solutions. The `archive` size is maintained by by employing crowding distance strategy [7]. In order to accommodate the new nondominated solutions in each generation, the excess nondominated solutions in the `archive` are removed by using crowding distance strategy. The execution of this strategy is described in subsequent sections (refer section5.3.3.6). However, population individuals having identical fitness values can have similar crowded distance values and increase in number of these individuals affects the diversity among nondominated solutions present in the `archive`. Therefore, to exhibit diversity among solutions, a polynomial mutation operation is performed on the best fitness individual (i.e., q_s^{lsed}) in each generation of MOMRFD method. In the next step, the mutated individual (q_s^{mlsed}) and best fitness individual (q_s^{lsed}) are compared, and the individual having the best fitness value is selected and stored in the `archive`. The procedure for applying polynomial mutation operator to a variable y in MOMRFD is given below [52].

(i) Generate a random number u between 0 and 1.

(ii) Generate a parameter δ_q,

$$\delta_q = \begin{cases} [2u + (1-2u)(1-\delta)^{\eta_m+1}]^{\frac{1}{\eta_m+1}} - 1 & if\ u \leq 0.5 \\ 1 - [2(1-u) + 2(u-0.5)(1-\delta)^{\eta_m+1}]^{\frac{1}{\eta_m+1}} & otherewise \end{cases} \tag{5.20}$$

where $\delta = min[(y-y_l),(y_u-y)]/(y_u-y_l)$. Here, y_u and y_l denote the upper and lower limits of the variable y, respectively; η_m represents distribution index of polynomial

mutation and $\eta_m \in$ [20,100] [161].

(iii) The mutated value of the variable y is evaluated using $y' = y + \delta_q(y_u - y_l)$.

5.3.3 Improved Multiobjective Brain Storm Optimization (IMBSO)

Although, brain storm optimization searches for optimum solution by exploring both dominated and non-dominated solution spaces, to administer the search movement and to account for any uncertainties in search direction, a number of improvements are integrated within multiobjective brain storm optimization (MBSO) algorithm [88]. We consider the MBSO algorithm presented in [88] for evaluation and following key features are added to improve the MBSO algorithm.

- Incorporation of an efficient k-means++ algorithm as an initial seeding technique for k-means clustering algorithm during clustering step.
- Selection of optimal cluster centroids during population generation step using river formation dynamics (RFD) method.
- Generation of new individuals using adaptive Lévy mutation operator [162] to preserve diversity.
- Incorporation of two different constraint handling strategies for handling boundary constraints using random procedure [115] and for handling functional constraints using constrained-domination principle [7].

An efficient k-means++ algorithm is employed as an initial seeding technique for k-means clustering algorithm during clustering step of IMBSO algorithm to avoid inclusion of distinct ideas in similar clusters. With the currently available random probability evaluation during selection of cluster centroids as selected individuals, a fraction of problem instances escapes without adequate treatment. Therefore, to facilitate the process of selection of an individual from a pool of elite individuals to act as a parent for next generation in IMBSO

algorithm, the design parameters are tailored to fit a random probability distribution function generated by RFD to account for an appropriate outcome (optimal cluster centroids are chosen as selected individuals) and simultaneously exhibit low computational search cost. Moreover, from circuit design point of view, the IMBSO algorithm is developed to explore multi-dimensional performance space spanned by the design specifications to retrieve an optimal set of design variables (or parameters) subject to design constraints. The design constraints are handled by incorporating both random boundary constraint handling technique and constrained-domination principle based handling technique inside IMBSO algorithm.

5.3.3.1 Clustering Strategy

Clustering is a division of ideas (individuals) into similar groups. Each group, called a cluster, consists of ideas that are similar between themselves and are unrelated to ideas of other clusters. The strength of similarity (or homogeneity) within a group and the strength of difference between groups define the quality of clustering. Each idea in a cluster is closer (more similar) to a prototype that defines the cluster. Often a prototype is considered as a $centroid$ for ideas with continuous attributes, i.e., the average of all points in the cluster. The selection of initial cluster centroids is crucial in iterative clustering techniques, like k-means clustering technique, as it has a direct impact on the formation of final clusters. Since k-means is a widely used clustering technique, it is also incorporated as a clustering strategy to classify individuals into separate clusters in MBSO algorithm. However, the generation of clusters using k-means algorithm depends on initial random assignments of cluster centroids. This limits the generation of well separated centroids and does not ensure a global minimum of variance.

Choosing the number of clusters is crucial to the separation of ideas in different clusters. As the number of clusters are kept fixed during each iteration of k-means clustering, distinct idea(s) (individuals) may be grouped with other ideas having similar attributes, and the characteristics of these ideas may get lost during analysis. These disadvantages compel researchers to employ other clustering techniques instead of k-means algorithm. Different

clustering strategies are presented in literature as clustering strategies for MBSO algorithm, *i.e.*, Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise (DBSCAN) strategy [163] and a simple random clustering strategy [164]. Although k-means clustering algorithm suffers from the aforementioned disadvantages, the simplicity and speed of the algorithm are quite appealing in practice [165]. Further, the disadvantages (initialization of cluster centers) of k-means algorithm can be addressed by using a simple, randomized seeding technique, called k-means$++$ [165]. Therefore, we choose to introduce k-means$++$ algorithm as an augmentation to k-means algorithm for optimal clustering in IMBSO framework. The k-means algorithm selects k centers in a single iteration according to uniform distribution of ideas (individuals), whereas k-means$++$ algorithm selects one individual in each iteration according to a non-uniform distribution using k iterations, which provides a gain over random initialization by constantly updating the non-uniform selection of individuals in each iteration. k-means$++$ selects initial centers of clusters using following steps [165],

(i) Select one individual as the centroid uniformly at random among the individuals in each cluster.

(ii) Evaluate Euclidean distance of each individual from the nearest selected center.

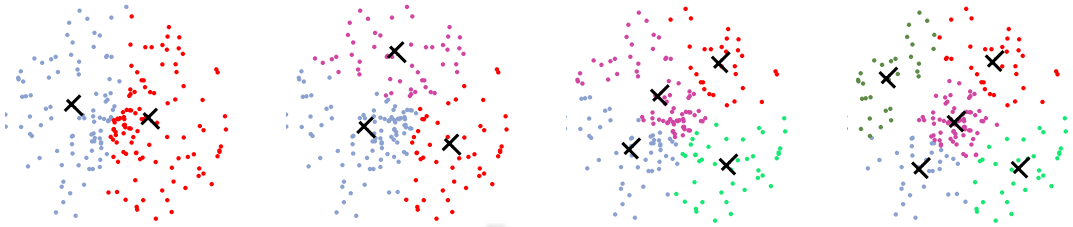
(iii) Randomly choose another individual as the new centroid using a weighted probability distribution where each individual is chosen with the probability proportional to squared distance.

(iv) Repeat steps 2 and 3 until k centroids are selected.

(v) Once initial centroids are chosen, standard k-means clustering algorithm can be applied.

The k-means$++$ seeding technique avoid additional degrees of freedom by limiting introduction of any new parameters (such as number of runs for k-means$++$) in the clustering algorithm. By clustering along distinct ideas (individuals), the technique introduces a measure of

distance between cluster centroids augmenting k-means clustering technique for evaluation of the minimum distance between cluster centroid and cluster individuals.



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

Figure 5.8: Initial selection of cluster centroids using k-means++ clustering technique after dividing individuals into (a) two clusters, (b) three clusters, (c) four clusters, (d) five clusters.

In view of this, a number of clusters (two to five) are generated for 100 random generated individuals using k-means++ seeding technique as shown in Figure 5.8 (without applying k-means clustering technique). It can be observed from Figure 5.8 that the clustered individuals are not bound to a single point (separate ideas) and the centroids are generated successfully near to the center individual using k-means++ clustering technique. Once initial k centroids are selected using k-means++ algorithm, k-means clustering technique is employed to cluster the individuals into k clusters and the procedure is followed to generate P_{elite} and P_{normal}.

5.3.3.2 Population Generation

In MBSO framework, new individual generation process is performed by generating individuals from P_{elite} or P_{normal} or $\mathrm{P}_{archive}$ based on three probabilities `p_one`, `p_two` and `p_three`, respectively. However, in [163] and [164], the authors make use of additional probabilities to control the exploitation region for the selection of an elite individual ($\mathbf{x}_{selected}$) and the subsequent generation of new individual ($\mathbf{x}_{new}$) using different mutation strategies (i.e., differential evolution mutation and open probability mutation). As the decision of selecting an individual from P_{elite} set is emphasized on random probability, the decision seems to lack exploitation of global information of entire population. Accordingly to make use of the global information and further improvement of evolutionary diversity of MBSO algorithm, the probabilistic decision making scheme of RFD is employed in this section. During evaluation of

new population, for selecting the cluster centroid ($\mathbf{x}_{selected}$), additional probability calculation is replaced by evaluating the centroid having best fitness value ($\mathbf{x}_{selected}$) among individuals within cluster(s) using RFD. Further, an adaptive mutation operator (adaptive Levy mutation operator [162]) is incorporated in IMBSO framework for improved performance. The complete evaluation procedure of IMBSO framework is described in Algorithm 19.

5.3.3.3 Selection of cluster centroid using RFD scheme

MBSO algorithm updates the cluster centroids in each iteration which shows a trend of moving towards better solutions. In this regard, we propose to incorporate RFD scheme inside IMBSO framework. Here, the purpose behind this is to select an optimal centroid among individuals in one cluster (`random(0,1)` $<$ `p_three`) or multiple (two) clusters (`random(0,1)` $>$ `p_three`).

For selecting an optimal centroid in one cluster, the probability (Pr) (as described in (4.8)) of selecting the next centroid in a cluster is estimated by evaluating the gradient (D_s) between current centroid and previous centroid of the cluster, or considering the gradients between current centroid and all nearest elite individuals in the current cluster. However, it can also be evaluated by considering gradients of all possible choices at the current cluster centroid instead of taking a single gradient. The amount of erosion introduced by RFD is considered as a penalty to the fitness (i.e., optimal cluster centroid) selection and it is estimated by the cost being paid in selecting the next best fitness (centroid). During evaluation of cluster centroid, each individual is assumed as node and the current fitness values of individuals represent different altitude values (Alt). The Euclidean distance between nearest individuals correspond to the weight (w) of edge between nodes (individuals). Such implementation of RFD administers the evaluation of a cluster centroid in optimal direction (best cluster centroid evaluation). The process is continued with a number of water drops starting from one centroid and performing random probabilistic movement to follow different paths (selecting different individuals each time). This process of evaluation continues till the all individuals in the clusters are analyzed or an iteration limit is reached. Finally, all possible evaluation paths are

Algorithm 19: Procedure for the proposed improved multiobjective brain storm optimization (IMBSO)

Input : Bounds of variable vector $\mathbf{x}$, N, `max_gen`, M
Output : $\mathrm{P}_{archive}$

Generate initial population of size N for variable vector $\mathbf{x}$;
$\mathrm{P}^1_{archive} \leftarrow \phi$; ▷ Initially empty $\mathrm{P}_{archive}$
$\mathrm{P}^1_{elite} \leftarrow \phi, \mathrm{P}_{normal} \leftarrow \phi$; ▷ P_{elite}: Set of elite individuals, P_{normal}: Set of normal individuals
$\mathbf{F} \leftarrow f(\mathbf{x})$; ▷ evaluate fitness for each objective M
if $\mathbf{x} \npreceq \mathbf{y} \; \forall \, \mathbf{y} \in \mathrm{P}^1_{archive}$ **then**
  $\mathrm{P}^1_{archive} \leftarrow \{\mathbf{y} \in \mathrm{P}^1_{archive} \mid \mathbf{y} \nprec \mathbf{x}\}$; ▷ Add nondominated points to $\mathrm{P}_{\mathrm{archive}}$ and remove points dominated by $\mathbf{x}$
  $\mathrm{P}^1_{archive} \leftarrow \mathrm{P}^1_{archive} \cup \mathbf{F}$;
end if
for $t \rightarrow 2$ to `max_gen` **do**
  Initialize cluster centroids, $c_1, c_2, \ldots, c_k \in \mathbb{R}$ of clusters $C_1, C_2, \ldots C_k$, respectively using k-means++;
  Cluster variable vector $\mathbf{x}$ into k (> 1) clusters for M objectives; ▷ Clustering analysis starts using $\mathrm{k-means}$
  $p \leftarrow 1$;
  while $p < N - 1$ **do**
    for $i \leftarrow 1$ to N **do**
      for $j \leftarrow 1$ to k **do**
        $l_{iq} \leftarrow \arg\min_j ||x_i - c_j||^2, q = j, \; q \in k$;
      end for
      $Gr_q \leftarrow Gr_q \cup x_i$;
    end for
    for $j \leftarrow 1$ to k **do**
      $c_j \leftarrow \mathrm{mean}(Gr_j)$;
    end for
    $p \leftarrow p + 1$
  end while ▷ Clustering ends using $\mathrm{k-means}$
  For each objective M, choose one cluster as `Elite_cluster` having the best fitness value;
  if $\mathbf{x} \in$ `Elite_cluster` **then**
    $\mathrm{P}^t_{elite} \leftarrow \mathrm{P}^t_{elite} \cup \mathbf{F}$;
  else
    $\mathrm{P}^t_{normal} \leftarrow \mathrm{P}^t_{normal} \cup \mathbf{F}$;
  end if
  for $n \rightarrow 1$ to N **do**
    if `random(0,1)` $<$ `p_one` **then**
      if `random(0,1)` $<$ `p_two` **then**
        if `random(0,1)` $<$ `p_three` **then**
          Randomly select an `Elite_cluster` and a cluster centroid;
          $\mathbf{x}_{selected} \leftarrow$`findCentroidUsingRFD (Elite_cluster)`;
        else
          Randomly select two clusters, C_i and C_j, $i, j \in k$;
          $C \leftarrow C_i \cup C_j$;
          $c \leftarrow$ `chooseRandomCentroid(`C`)`;
          $\mathbf{x}_{selected} \leftarrow$ `findCentroidUsingRFD(`C`)`;
        end if
      else
        $\mathbf{x}_{selected} \leftarrow$ `chooseRandom` $(\mathrm{P}^t_{archive})$;
      end if
    else
      $\mathbf{x}_{selected} \leftarrow$ `random` $(L_{\mathbf{x}}, U_{\mathbf{x}})$; ▷ $L_{\mathbf{x}}, U_{\mathbf{x}}$ denote lower and upper bounds of variable vector $\mathbf{x}$
    end if
    $\mathbf{x}_{new} \leftarrow$ `AdaptiveLevyMutation` $(\mathbf{x}_{selected})$;
    $\mathbf{x} \leftarrow$ `Selection` $(\mathbf{x}_{selected}, \mathbf{x}_{new})$;
  end for
  $\mathbf{F} \leftarrow f(\mathbf{x})$;
  $j \leftarrow 1$
  while $|\mathrm{P}^t_{archive}| + |\mathbf{F}_j| \leq N$ **do**
    `AssignCrowdingDistance`$(\mathbf{F}_j)$;
    $\mathrm{P}^t_{archive} \leftarrow \mathrm{P}^t_{archive} \cup \mathbf{F}_j$;
    $j \leftarrow j + 1$;
    `QuickSort`$(\mathbf{F}_j)$;
  end while
end for

analyzed and individual having the best fitness (minimum value) is represented as centroid of the cluster and is chosen as $\mathbf{x}_{selected}$.

Algorithm 20: Procedure for `findCentroidUsingRFD()`

Input : Individuals of a single cluster, $\mathbf{x} \in C_i$, $i \in k$, or individuals of a combined cluster, $\mathbf{x} \in C, C = C_i \cup C_j, i, j \in k$

Output : $\mathbf{x}_{selected}$

1: **for** $p \leftarrow 1$ to N_C **do**
2: $sum \leftarrow \phi$
3: **for** $j \leftarrow 1$ to $numOfDrops$ **do**
4: $total\_erosion_j^p \leftarrow \phi$
5: **while** $\mathbf{x}_j^p$ *is not covered* **do**
6: $r \leftarrow$ `random(0,1)`
7: **for** $k \leftarrow 1$ to $\mathbf{x}_{neighbor}^p$ **do**
8: **if** $r < Pr(\mathbf{x}_j^p, \mathbf{x}_{neighbor}^p)$ **then**
9: *break*
10: **end if**
11: **end for**
12: $\mathbf{x}_{next}^p \leftarrow \mathbf{x}_j^k$ ▷ find the next individual
13: $\mathbf{x}_{neighbor}^p \leftarrow$ `FindNeighbors`$(\mathbf{x}_{next}^p)$ ▷ find neighbors of the next individual
14: $total\_erosion_j^p \leftarrow total\_erosion_j^p + Erosion(\mathbf{x}_j^p, \mathbf{x}_{next}^p)$ ▷ update erosion value of p^{th} individual
15: **end while**
16: **if** $|\mathbf{x}_j^p| == \phi \,||\, |\mathbf{x}_j^p| == |\mathbf{x}_{next}^p|$ **then**
17: $\mathbf{x}_j^p \leftarrow \mathbf{x}_j^p + \epsilon$ ▷ $\epsilon \leftarrow (total\_erosion^p)/(N_C - 1)$ denotes sedimentation parameter
18: **end if**
19: $\mathbf{x}_j^p \leftarrow \mathbf{x}_j^p - total\_erosion^p$
20: $sum \leftarrow sum + \mathbf{x}_j^p$
21: **end for**
22: $\mathbf{x}^p \leftarrow (sum)/(numOfDrops)$
23: **if** $\mathbf{x}^{p+1} \not\preceq \mathbf{x}^p \; \forall \; \mathbf{x}^p \in \mathrm{C}$ **then**
24: $\mathbf{x}_{optimal} \leftarrow \mathbf{x}^p$
25: **end if**
26: **end for**

For selecting multiple clusters, a higher random number is generated, which is greater than `p_three` (step 34 of Algorithm 19, i.e., if condition fails). In this manner, two clusters are selected at random instead of one for further selection of the individual. The two clusters are combined to form a single cluster and accordingly, the optimal cluster centroid (having best fitness value in the combined cluster) is evaluated using RFD scheme and is chosen as $\mathbf{x}_{selected}$. Whenever necessary, other evaluation steps are employed similar to the algorithmic steps of MBSO algorithm. Once an individual is selected, i.e., $\mathbf{x}_{selected}$, mutation operation is carried out followed by selection operation for the generation of new population.

The process of finding an optimal centroid using RFD scheme can be considered as a process of minimization, where the individual having the minimum fitness is chosen as $\mathbf{x}_{selected}$. The entire process of minimization using RFD is described in Algorithm 20. The process of

finding an optimal centroid starts by considering the decision variables, **x** as individuals in cluster C. During evaluation, the probability Pr, and decreasing gradient D_s among individuals and their neighbors are evaluated using (4.8) and (4.9), respectively. The neighboring individual having the maximum probability is selected as the next individual ($\mathbf{x}_{next}$) and the fitness value (variable value) is reduced by the amount of erosion ($Erosion$) between the two individuals. The amount of erosion between two individuals, **x** and $\mathbf{x}_{next}$ is evaluated as follows [85],

$$Erosion(\mathbf{x}, \mathbf{x}_{next}) = \frac{\alpha}{(N_C - 1) \times (d)} \times D_s(\mathbf{x}, \mathbf{x}_{next}) \tag{5.21}$$

where $Erosion(\mathbf{x}, \mathbf{x}_{next})$ denotes the amount of erosion, when a drop travels from individual **x** to individual $\mathbf{x}_{next}$; $\alpha = \texttt{rand}(0, 1)$ denotes the erosion parameter; N_C represents the number of individuals in cluster C and $d = 25$ is the number of drops used in the algorithm. The number of drops, d, is considered as per the analysis performed in [135]. As excess erosion may develop flat surfaces across landscapes (i.e., the variable values of individuals may become equal or close to zero) making the gradient evaluation insignificant, it is necessary to increase the values of decision variables by depositing sediments, ϵ, to avoid any premature convergence. This movement of water drop continues until all individuals in the cluster are covered (or an iteration limit is reached) and the variable value, **x**, is updated each time a drop travels to a neighboring individual. Accordingly, a number of drops are employed to follow different paths by covering separate individuals. In the end, all possible potential paths are analyzed and the final fitness value (variable value) of an individual is evaluated by taking the average. The final variable values of all individuals in the cluster are evaluated and the best fitness value (optimal cluster centroid, $\mathbf{x}_{selected}$) is selected by Pareto dominance.

5.3.3.4 Adaptive Lévy Mutation

A number of mutation operators are incorporated in literature to perform mutation operation in MBSO framework, i.e., Gaussian mutation [88], Cauchy mutation [88], differential evolution mutation [163] and adaptive open probability mutation [164] operators. However,

by employing Lévy probability distribution, the mutation operator can be extended and generalized for population generation in IMBSO framework [162]. Both Gaussian and Cauchy probability distribution can be represented as special cases of Lévy probability distribution. Because of power law in the tail region and flexibility in modifying the probability density to adjust variation in the mutation, Lévy probability distribution has been employed in a number of real life applications [156]. In this regard, we propose to incorporate an adaptive Lévy mutation operator for population generation. After incorporating Lévy mutation operator, (2.6) can be rewritten as,

$$\begin{aligned} \mathbf{x}_{new} =& \mathbf{x}_{selected} + \xi L_{\alpha}(s), \\ \xi =& \texttt{logsig}\left(\frac{T-t}{2K}\right) \times rand(), \end{aligned} \tag{5.22}$$

where $L_{\alpha}(s) \sim s^{-1-\alpha}$, denotes a random number generated using Lévy distribution with the parameter α, such that $0 < \alpha < 2$ and s represents the distance parameter [166]. The parameter α controls the shape of distribution, $i.e.$, for α=1, the Lévy distribution becomes Cauchy distribution and for α=2, the distribution becomes Gaussian distribution. Lévy mutation operator can use a number of values of α to generate different random numbers by dynamically updating (5.22). This can lead to a large variation in step size and the subsequent generation of distinct individuals (preservation of diversity by coming out of local optima).

5.3.3.5 Selection

In IMBSO algorithm, selection operator is used to improve the average quality of population by choosing better individuals to be copied into the next generation and the selection operation is based on Pareto dominance similar to MBSO algorithm [88]. The Pareto dominance principle is based on following definition [7].

Definition 1. *A solution x is said to dominate another solution y (denoted as $x \prec y$), if both conditions 1 and 2 are true.*

1. The solution x is no worse than solution y in all objectives, or $f_i(x) \not\succ f_i(y)$, $\forall\ i = 1, 2, \ldots, M$, where M denotes the number of objective functions.

2. The solution x strictly dominates solution y in at least one objective, or $f_i(x) < f_i(y)$, for at least one $i \in \{1, 2, \ldots, M\}$.

During selection operation, the next individual is chosen between the selected individual ($\mathbf{x}_{selected}$) and the mutated individual ($\mathbf{x}_{new}$) by following Pareto dominance conditions described in Definition 1. The selection rules are as follows,

(i) If $\mathbf{x}_{selected}$ dominates $\mathbf{x}_{new}$, then $\mathbf{x}_{selected}$ survives.

(ii) If $\mathbf{x}_{new}$ dominates $\mathbf{x}_{selected}$, then $\mathbf{x}_{new}$ survives.

(iii) If any of the Pareto dominance conditions are violated, the next individual is chosen randomly between $\mathbf{x}_{selected}$ and $\mathbf{x}_{new}$.

5.3.3.6 Updating Global Archive ($\mathrm{P}_{archive}$)

In IMBSO framework, $\mathrm{P}_{archive}$ is updated by new nondominated solutions (selected individual) in each generation after selection process. Individuals having better nondominated solutions are kept by limiting the size of $\mathrm{P}_{archive}$ using crowding distance operator followed by niche preservation. At first, all individuals in $\mathrm{P}_{archive}$ are set to zero crowding distance values and are sorted in ascending order using quick sort. The crowding distance of individuals at the first and last positions are updated to infinity and the crowding distance of other individuals are evaluated as follows [7],

$$d(i) = d(i) + \frac{f_j(i+1) - f_j(i-1)}{f_j^{max} - f_j^{min}}, \quad i = 2, 3, \ldots, N-1, \quad j = 1, 2, \ldots, M, \tag{5}$$

where $d(i)$ denotes the crowding distance of i^{th} individual; $f_j(i)$ represents the i^th individual function value on the j^{th} target (i.e., j^th objective); f_j^{max} and f_j^{min} denote the maximum and minimum of all individuals in the function value of the j^th target (j^{th} objective), respectively; N and M denote the number of individuals in the $\mathrm{P}_{archive}$ and the number of objectives, respectively. The crowding distance of all individuals is evaluated once $\mathrm{P}_{archive}$ is full and the individual having the smallest crowded distance is excluded from $\mathrm{P}_{archive}$. While evaluating

crowding distance of individuals in $\mathrm{P}_{archive}$, individuals having identical fitness values are tagged with similar crowding distance. Increase in number of these individuals with identical fitness values affects the diversity of Pareto set. Therefore, it is necessary to evaluate the niche count for each individual in $\mathrm{P}_{archive}$. The niche count is evaluated by counting number of points in the $\mathrm{P}_{archive}$ within a certain distance from an individual and the individual with the least niche count is preserved [167].

5.4 Design Examples

5.4.1 Two-stage Operational Amplifier

The design of operational amplifiers are represented as the fundamental analysis of analog circuit design. A case study of a two stage operational amplifier is presented as shown in Figure 5.9 to demonstrate the tradeoff between two competing performance specifications, i.e., DC gain (A_v) and power consumption (P_{DC}). The two-stage operational amplifier circuit is designed in 0.18 μm CMOS process technology using Cadence Virtuoso [168]. The closed-form expressions of A_v and P_{DC} are approximated as follows [142],

$$A_v = \frac{g_{m1}g_{m6}}{(g_{ds2}+g_{ds4})(g_{ds6}+g_{ds7})},\ P_{DC} = v_{dd}(I_1+I_2+I_3) \tag{5.23}$$

where g_m denotes the transconductance, $g_{ds} = 1/r_o$ represents the reciprocal of output impedance. The reference current I_{bias} is kept at I_{bias}= I_3=I_1 + I_2. The corresponding transconductances of different transistors are approximated using the closed form expressions as described in [142], $i.e.$, in terms of width (W) and length (L) of the transistors. The details of design parameters considered during evaluation is listed in Table 5.5 and, all boundary and functional design constraints are summarized in Table 5.6. The supply voltage is kept constant at $1.8V$ for suitable operation of transistors. Before starting the process of optimization, channel lengths of $0.18\mu m$ are selected and inversion coefficient is kept at below unity ($\simeq 0.8$) for M1 and M2 transistors.

However, other transistors are allowed to operate at the strong inversion side to minimize

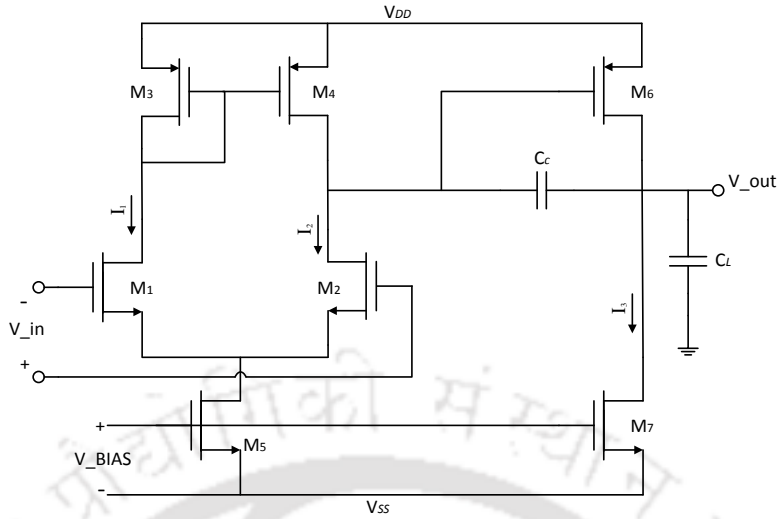


Figure 5.9: Two-stage operational amplifier circuit [142].

Table 5.5: Two-stage operational amplifier design parameters and ranges.

Variables	Description	Lower bound	Upper bound
W_1, W_2 [μm]	Width of M1 and M2	0.48	0.9
W_3, W_4 [μm]	Width of M3 and M4	2.11	2.7
W_6 [μm]	Width of M6	30.1	38.88
I_{bias} [μA]	bias current	16	20

their V_{DS}^{sat} or drain thermal-noise current inputs. The load capacitance C_{load} is kept at $10\ pF$. To ensure stable operation of the amplifier, miller capacitance (C_c) and load capacitance are bounded by the relation, $C_c > 2.2(g_{m2}/g_{m6})C_{load}$. Since the performance specifications of two-stage amplifier circuit, i.e., DC gain and power consumption compete each other by progressing in maximum and minimum directions, the sign of DC gain value is inverted during multiobjective minimization. This method of inverting the sign analyzes DC gain value which is to be maximized by converting it into a minimization problem.

Table 5.6: Two-stage operational amplifier constraints and specifications.

Specifications	Description	Constraints
ϕ [degree]	Phase margin	≥ 60
P_{max} [mW]	Max. power consumption	≤ 300
SR [A/μF]	Slew rate	≥ 20
$ICMR$ [V]	Input common mode range	$[0.8, 1.6]$
GBW [MHz]	Gain bandwidth product	≤ 30

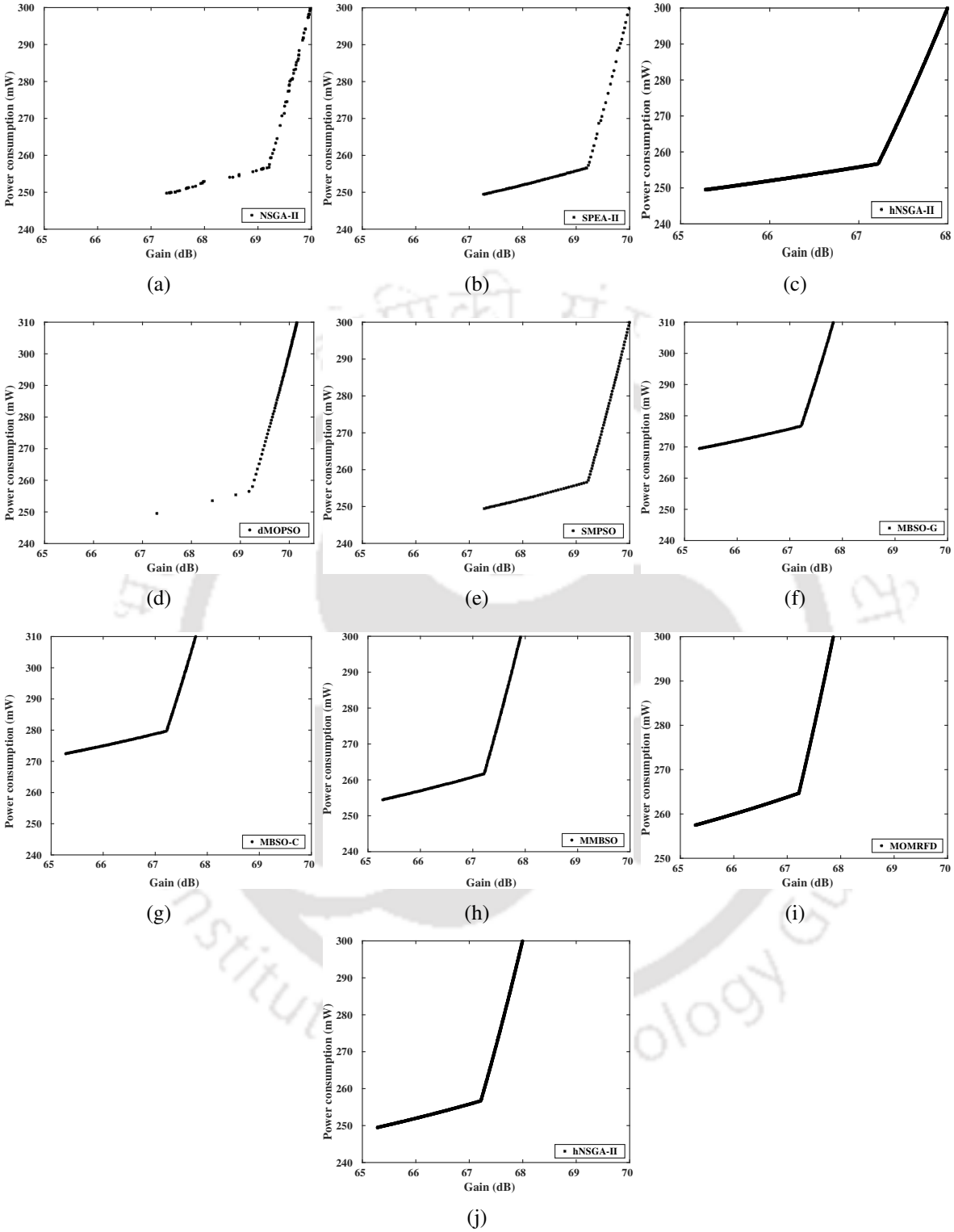


Figure 5.10: Plot of nondominated fronts for gain (max) and power consumption (min) of two-stage operational amplifier by (a) NSGA-II, (b) SPEA-II, (c) hNSGA-II, (d) dMOPSO, (e) SMPSO, (f) MBSO-G, (g) MBSO-C, (h) MMBSO, (i) MOMRFD, (j) IMBSO.

It can be observed from Figure 5.10 that DC gain (A_v) maximizes sub-linearly for a fixed biasing at an expense of core $256mW$ power consumption (P_{DC}) under several design constraints listed in Table 5.6. This sub-linear plot of Pareto front (demonstrating tradeoff between DC gain and power consumption) is characterized by two different regions. In the first region, where $A_v < 67.5$ dB, A_v increase at a slower pace with increase in P_{DC}. However, in the second region, small increase in A_v shows a significant increase in P_{DC}. Further, to demonstrate the effectiveness of proposed multiobjective optimization algorithms, the amplifier circuit is optimized for maximum A_v at the expense of P_{DC} using several optimization techniques. It can be observed from Figure 5.10 that a smooth distribution of final Pareto optimal solutions is obtained using proposed hNSGA-II, MOMRFD and IMBSO algorithms over a finite search space.

5.4.2 Folded Cascode Amplifier

In the second design example, folded cascode topology is exhibited in circuit design process to improve gain and input common mode range at the expense of power consumption and GBW. In this section, a case study of folded cascode operational amplifier circuit is designed with 0.18μm CMOS process technology using Cadence Virtuoso as shown in Figure 5.11. During design, load capacitance C_L is kept at 5 pF and proper differential operation and current biasing is ensured by the matching relations (e.g., M1≡M2, M3≡M4, M5≡M7, etc.) between the transistors. The analytical form of performance specifications are approximated as follows [142].

$$\begin{aligned} A_v &= g_{m2}R_{out} \\ R_{out} &= (g_{m8}r_{ds8}r_{ds6}) \parallel [g_{m10}r_{ds10}(r_{ds2} \parallel r_{ds4})] \\ P_{DC} &= v_{dd}(I_1 + I_2 + I_3 + I_4),\ GBW = \frac{g_{m2}}{2\pi C_L},\ SR = \frac{I_2}{C_L} \end{aligned} \tag{5.24}$$

where A_v, R_{out}, P_{DC}, GBW and SR denote DC gain, output impedance, power consumption, gain bandwidth product and slew rate of folded cascode amplifier circuit, respectively; g_m and r_{ds} denote the transconductance and drain to source resistance of transistor, respectively.

It is to be noted that the currents are assumed as, I_1=I_2=I_{bias}, I_3=I_4=0.7I_{bias}. The design parameters and ranges are listed in Table 5.7. Here we choose to formulate the design problem as a multiobjective optimization problem to maximize A_v and minimize P_{DC} subject to a set of constraints listed in Table 5.8.

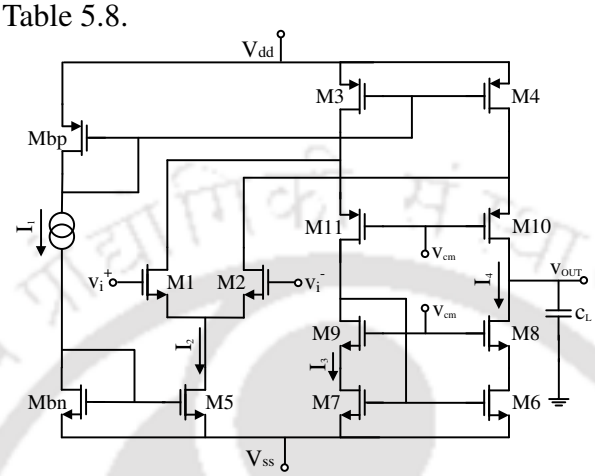


Figure 5.11: Folded Cascode Operational Amplifier circuit

Table 5.7: Folded cascode operational amplifier design parameters and ranges.

Variables	Description	Lower bound	Upper bound
W_1, W_2 [μm]	Width of M1 and M2	0.24	4.0
W_3, W_4 [μm]	Width of M3 and M4	0.24	4.1
W_5, W_7 [μm]	Width of M5 and M7	0.24	5.3
W_6 [μm]	Width of M6	0.24	101.88
W_8 [μm]	Width of M8	0.35	101.88
W_{10} [μm]	Width of M10	0.21	120.88
I_{bias} [mA]	Bias current	1e-03	2.5

Table 5.8: Folded cascode operational amplifier constraints and specifications.

Specifications	Description	Constraints
A_v [V]	DC gain	≥ 55
ϕ [degree]	Phase margin	≥ 60
P_{DC} [μW]	Power consumption	≤ 200
SR [V/μS]	Slew rate	≥ 1
V_{swing} [V]	Output voltage swing	≥ 1.2
GBW [MHz]	Gain bandwidth product	≥ 2

To demonstrate the tradeoff among A_v and P_{DC} of folded cascode amplifier, nondominated solutions are evaluated using peer algorithms and three proposed multiobjective optimization algorithms. The corresponding Pareto fronts are plotted and shown in Figure 5.12. It can be observed from Figure 5.12 that DC gain can be maximized at the expense of power

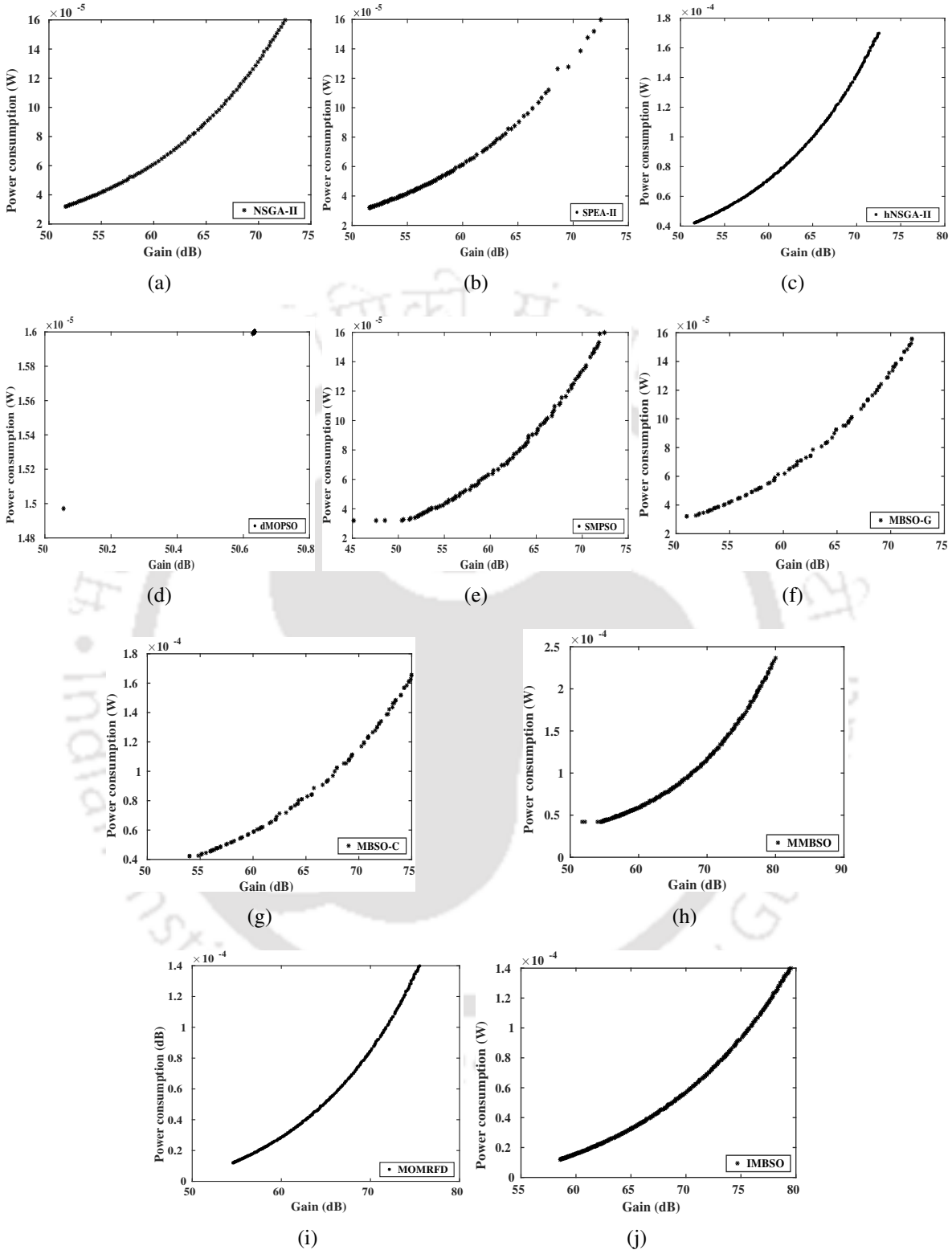


Figure 5.12: Plot of nondominated fronts for gain (max) and power consumption (min) of folded cascode amplifier by (a) NSGA-II, (b) SPEA-II, (c) *h*NSGA-II, (d) dMOPSO, (e) SMPSO, (f) MBSO-G, (g) MBSO-C, (h) MMBSO, (i) MOMRFD, (j) IMBSO.

consumption under 200 μW subject to the design constraints listed in Table 5.8. The sign of DC gain value is inverted during implementation to convert the maximization problem into the minimization problem. Further, it can be observed from Pareto fronts generated by peer algorithms and the proposed algorithms that the peer algorithms fail to generate a continuous Pareto front except NSGA-II and the proposed algorithms, i.e., hNSGA-II, MOMRFD and IMBSO. However, IMBSO algorithm generates nondominated solutions of A_v at the expense of less P_{DC} than NSGA-II and other proposed methods.

5.4.3 Low Noise Amplifier

In this section, a case study of source degenerate cascode low noise amplifier (LNA) circuit is considered as shown in Figure 5.13 to demonstrate the applicability of the proposed multiobjective optimization algorithms. LNA is designed using 0.18μm CMOS technology with 1.8V power supply centered around 2.4GHz frequency using Cadence Virtuoso [168]. The design of LNA is a relatively complex process as there exists tradeoffs among a number of performance specifications, such as noise figure (NF), gain, power consumption, etc.

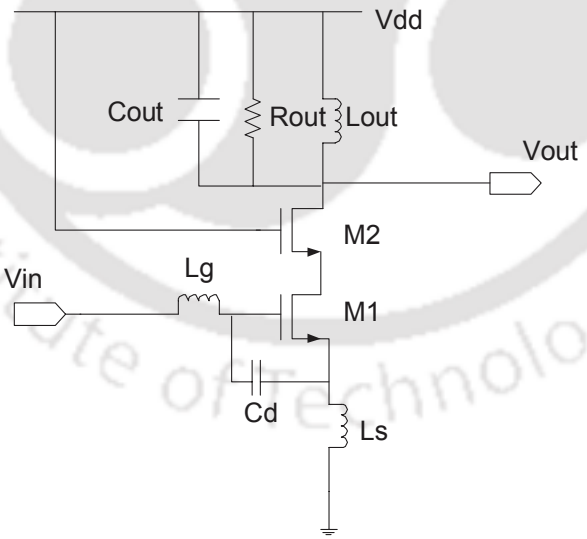


Figure 5.13: Low noise amplifier circuit [147].

During design of LNA, the NF is kept at minimum value ($i.e.$, ≤ 2.5 dB) for a drain current of $1m$A and a moderately high quality factor ($Q \in [3, 5]$) at an expense of $8m$W power consumption. As gain is crucial to the design process, sacrificing gain to minimize NF results

in the transmission of noise to other blocks of the receiver system during communication process. In this regard, the design problem is transformed into a constrained-aware multiobjective optimization problem with an objective to minimize NF and to maximize gain (S21) of LNA subject to a set of design constraints. Table 5.9 summarizes the ranges of variables specified during optimization of LNA. All design constraints and specifications are listed in Table 5.10. As the proposed multiobjective algorithms are developed for minimization of objective functions, during realization, the sign of gain value is inverted to convert the maximization problem into minimization problem. Similar to the design examples discussed previously, the closed-form expressions of NF and other design constraints are approximated on the design as reported in [147].

Table 5.9: LNA design parameters and ranges.

Variables	Description	Lower bound	Upper bound
W_1, W_2 [μm]	Width of M1 and M2	1	100
I_d[mA]	Forward drain current	0.1	4.5

Table 5.10: LNA constraints and specifications.*Ratio of intrinsic gate capacitance of $M1$ to total capacitance ($C_{tot} = C_{gs} + C_{ext} + C_p$).

Specifications	Description	Constraints
$S11$[dB]	Input reflection coefficient @ 2.4GHz	≤ -20
$S12$[dB]	Reverse isolation @ 2.4GHz	≤ -50
$S21$[dB]	Forward power gain @ 2.4GHz	≥ 10
$S22$[dB]	Output reflection coefficient @ 2.4GHz	≤ -0.9
NF_{min}[dB]	Min. noise figure @ 2.4GHz	≤ 2.5
P_{max}[mW]	Max. power consumption @ 2.4GHz	≤ 8
f_{cf}[GHz]	Center frequency	$[2.1, 2.7]$
C_{gs}/C_{tot}	Capacitance ratio*	≤ 1
Q	Quality factor	$[3, 5]$
Z_{out}^{min}[Ω]	Min. output impedance (real part)	≥ 50

Figure 5.14 plots the distributions of final nondominated solutions found in several optimization techniques including the proposed algorithms for LNA optimization. It can be observed from Figure 5.14 that the distribution of nondominated solutions in the Pareto front is quiet nonuniform for other optimization techniques except algorithms based on MBSO. Since the solutions are crowded in a corner of the Pareto front rather being distributed along

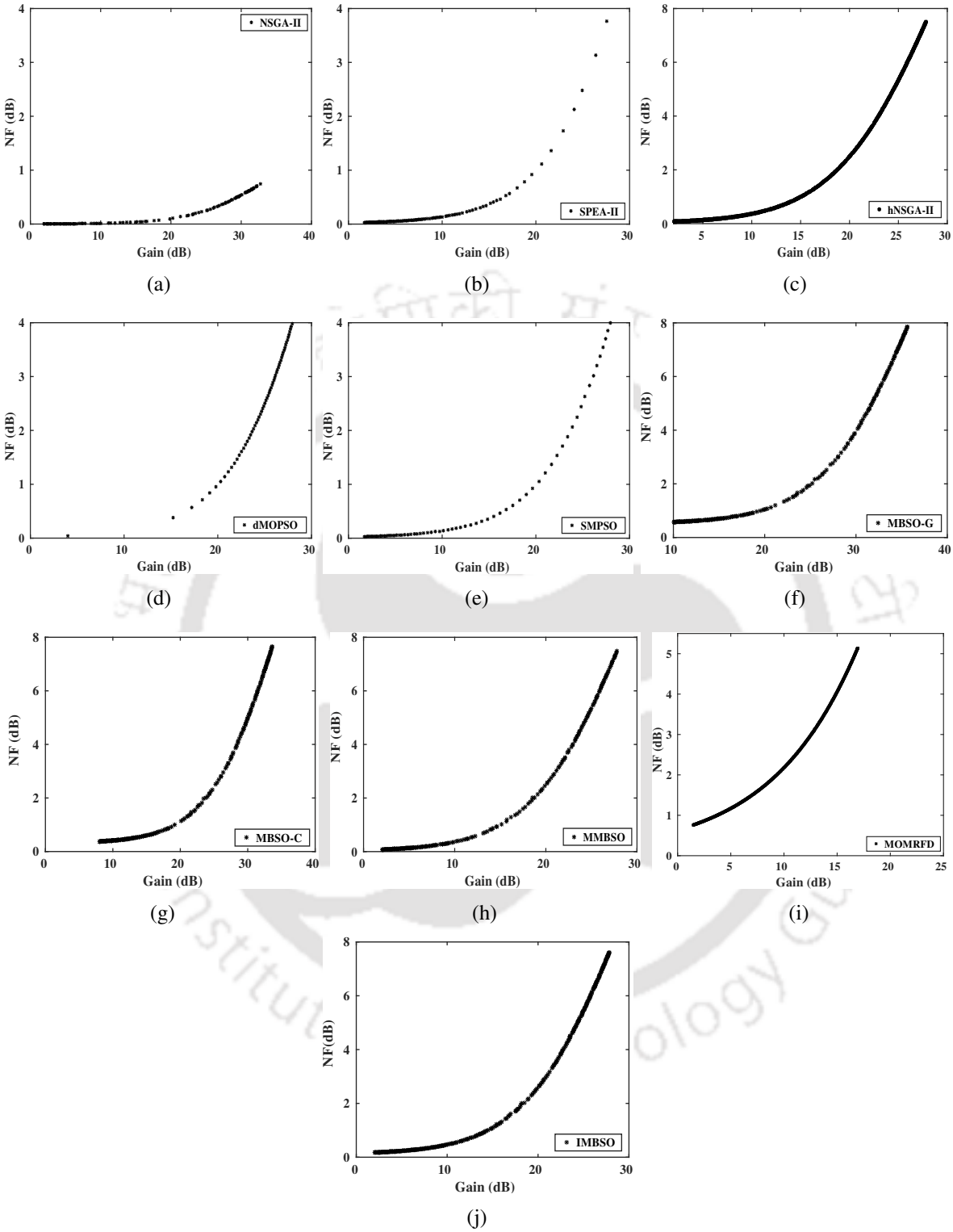


Figure 5.14: Plot of nondominated fronts for gain (max) and noise figure (min) of low noise amplifier by (a) NSGA-II, (b) SPEA-II, (c) hNSGA-II, (d) dMOPSO, (e) SMPSO, (f) MBSO-G, (g) MBSO-C, (h) MMBSO, (i) MOMRFD (j) IMBSO.

the front, it can be observed that the convergence is affected with the adoption of stringent design constraints. However, in the case of algorithms based on MBSO, both boundary and functional constraints are handled separately to avoid premature convergence and it is observed that a continuous distribution of solutions is obtained. Figure 5.14(j) represents an optimal Pareto front for minimum NF and maximum gain (S21 parameter) of LNA generated using IMBSO. It can be observed from this plot that with the increase in gain below 15dB, NF increases slowly and after that NF starts to increase significantly at high gain values. LNAs having high NF deteriorates the performance of a receiver system significantly by transmitting input signals coupled with large noise to other blocks. Therefore, the feasible region of operation is to operate at a minimum noise figure of 0.8dB at a moderate gain of 13.01dB with a power consumption of less than $8m$W.

5.5 Simulation settings and parameters

The proposed multiobjective optimization algorithms, i.e., hNSGA-II, MOMRFD and IMBSO are implemented by using C/C++ language and all simulation experiments are performed on a computer with Intel Xeon E5-2620 processor having 64GB of RAM under Linux environment. During experiments, the population size is kept at 200 to evolve for a maximum generation of 1000. The maximum size of Pareto set ($\mathrm{P}_{archive}$) is fixed at 100. For each experiment, 25 independent runs are performed to observe the success in function evaluation. During evaluation, the pre-determined probability values, $i.e.$, `p_one`, `p_two` and `p_three` are set to 0.99, 0.8 and 0.8 respectively. During k-means$++$, five clusters are generated in all experiments to have a minimum SSE (sum of squared error). Further, with aim of giving a complete overview on the performance of proposed multiobjective optimization algorithms in analog/RF circuit optimization, six different state-of-art multiobjective optimization techniques, $i.e.$, NSGA-II [7], SPEA-II [169], dMOPSO [81], SMPSO [80], MBSO-G [88], MBSO-C [88], MMBSO [163] are considered. For the purpose of comparison, we have used the authors implementation of all algorithms. The constraint handling techniques are imple-

mented inside MBSO and MMBSO framework to handle boundary and functional design constraints. The parameter settings of peer algorithms are listed in Table 5.11, where N, N_s, and A denote the population size, swarm size, and archive size, respectively. The maximum number of iterations and the maximum number of evaluations are denoted by I and E_{max}, respectively. Mutation and crossover probabilities are represented by p_m and p_c, respectively and, η_m and η_c denote distribution index of mutation and crossover, respectively. Polynomial mutation and SBX crossover operators are used in algorithms NSGA-II, SPEA-II. Other parameters, i.e., c_1, c_2, w, v_1 and v_2 represent the parameters used for updation of velocity in both dMOPSO, and SMPSO algorithms.

Table 5.11: Parameter settings of peer algorithms

Algorithm	Parameter settings
NSGA-II, hNSGA-II, MOMRFD	N = 200, E_{max} = 100000, β=256, $p_c = 0.9, p_m = 1.0, \eta_c = \eta_m = 20.0$
SPEA-II	N = 200, A = 100, E_{max} = 100000, $p_c = 0.9, p_m = 1.0, \eta_c = \eta_m = 20.0$
dMOPSO	N_s = 200, Max. age = 2, I = 250000, $c_1, c_2, w \in$ [0,1], v_1, v_2=-1.0
SMPSO	N_s = 200, A = 100, I = 2500, p_m = 1.0, η_m = 20.0, $c_1, c_2, w \in$ [0,1], v_1, v_2=-1.0
MBSO-G MBSO-C	N = 200, A = 100, I = 1000, p_one = 0.8, p_two = 0.6 and p_three = 0.9, K=40, μ = 0, σ= 1
MMBSO	N =200, A = 100, I = 1000, p_one = 0.99, p_two = p_three = 0.8, p_four = p_five = 0.2

5.6 Constraint Handling

To analyze the tradeoffs among performance specifications bounded by design constraints, we propose to introduce two constraint handling techniques to handle boundary constraints and functional constraints separately inside the proposed multiobjective algorithm frameworks. Boundary constraints limit the values of design parameters, where each parameter is bounded by lower and upper limits ($e.g.$, ICMR design parameter described in (1.2)). Functional constraints represent both equality and inequality constraints ($e.g.$, other constraints except ICMR described in (1.2)).

- **Handling Boundary Constraints :** During analysis of boundary constraints, it is essential to ensure that the design parameter values lie inside bounded region after evaluation in each generation. This can be achieved by replacing the design parameter values outside the bounded region with a random value within the boundaries (random constraint handling technique) [115], which can be evaluated as follows [115],

$$x_i^D = l^D + rand(0,1) \times (u^D - l^D), \tag{5.25}$$

where x_i^D denotes i^{th} variable having dimension D; $rand(0,1)$ returns a real value between 0 and 1 with uniform distribution; l^D and u^D denote the lower and upper bounds of D^{th} dimension, respectively.

- **Handling Constraint Functions :** For handling inequality constraints during circuit design optimization, constrained-domination principle [7] is employed to identify and rank feasible solutions according to their nondomination level. The constrained-dominance principle is based on the following definition.

Definition 2. *A solution x is said to constrained-dominate a solution y, if any of the following conditions is true [7].*

(i) Solution x is feasible and solution y is not feasible.

(ii) Both solutions x and y are infeasible, but solution x has a smaller overall constraint violation.

(iii) Both solutions x and y are feasible and solution x dominates solution y.

According to [7], the feasible solution has a better nondomination rank than infeasible solution while employing constrained-domination principle. Among feasible solutions, the nondomination level is decided based on the fitness values. Further, among infeasible solutions, the solution having less constraint violations has a better rank.

5.7 Experimental Results and Discussion

To measure the performance in terms of convergence towards optimal Pareto front and diversity of solutions along Pareto front, *Hypervolume* (HV) metric [154] is evaluated for all peer algorithms and the proposed methods. Both median and interquartile range (IQR) values of HV metric conducted over 25 independent runs are listed in Table 5.12 (IQR is reported within braces). HV is a fair measure of the maximum area covered by the nondominated solutions. Fronts with higher HV value represents better quality of optimized solutions in terms of diversity of solutions. It can be observed from Table 5.12 that proposed IMBSO algorithm achieves superior HV values in all three test circuits with statistical confidence in all three cases ('$+$' symbol denotes a confidence level of 95% and "$-$" symbol means no statistical confidence is found (i.e., p-value > 0.05)).

Table 5.12: Median and IQR of the HV metric for analog/RF circuit optimization problems. 'opamp' denotes operational amplifier ('$+$' symbol denotes a confidence level of 95%).

Algorithm	Two-stage opamp	Folded cascode opamp	Low noise amplifier
NSGA-II	0.6045 (4.12e-04)[+]	0.8371 (3.05e-04)[+]	0.6012 (3.12e-04)[−]
SPEA-II	0.6079 (6.10e-03)[+]	0.7312 (6.34e-03)[+]	0.6085 (2.70e-04)[−]
hNSGA-II	0.6828 (3.76e-04)[+]	0.8382 (2.06e-04)[+]	0.7478 (3.28e-04)[+]
dMOPSO	0.6081 (1.31e-04)[+]	0.6086 (2.04e-04)[+]	0.5912 (4.10e-04)[−]
SMPSO	0.8029 (3.40e-03)[+]	0.8200 (4.13e-03)[+]	0.7465 (1.10e-04)[+]
MBSO-G	0.9131 (3.20e-04)[+]	0.8011 (3.70e-04)[+]	0.8112 (2.29e-04)[+]
MBSO-C	0.9040 (2.10e-04)[+]	0.7962 (1.18e-04)[+]	0.8022 (8.21e-03)[+]
MMBSO	0.9198 (1.09e-04)[+]	0.8214 (3.73e-04)[+]	0.8736 (5.10e-03)[+]
MOMRFD	0.7018 (4.22e-04)[+]	**0.8491** (2.17e-04)[+]	0.7731 (1.03e-04)[+]
IMBSO	**0.9210** (2.42e-04)[+]	0.8472 (4.25e-04)[+]	**0.8812** (3.28e-04)[+]

5.7.1 Diversity Maintenance

Both NSGA-II and SPEA-II algorithms use crowding distance among the solutions in the selection process to maintain diversity among population. However, this crowding distance scheme becomes inefficient when two or more solutions share a common fitness and it is not always easy to generate a uniform distribution of Pareto optimal objective vectors using these

algorithms [44]. Therefore, the proposed hNSGA-II algorithm has inferior performance in terms of diversity among solutions as compared to MBSO based methods and the proposed MOMRFD and IMBSO methods. In case of both IMBSO and MOMRFD, crowding distance method is used only to maintain the nondominated solutions in archive and to preserve diversity, a niche count is performed to select diverse individuals.

Further, multiobjective optimization algorithms based on PSO lack in the generation of diverse individuals as these algorithms are often trapped in local optima and very sensitive to changes in constriction coefficients [45]. As mentioned in above section, the proposed IMBSO algorithm demonstrates superior HV values in all three test circuits. One of the reasons for this superior performance of IMBSO is the merging of clusters with diverse individuals to select the next individual for mutation and selection. The diversity among the individuals of different clusters naturally leads to the diversity in the population. At second, with the incorporation of RFD scheme to choose an optimal centroid during population generation step, each distinct individual is analyzed several times as RFD covers the entire search space to estimate the best fitness in one cluster or multiple clusters. This analysis reduces the chances of skipping any alternate ideas (individuals) and premature convergence. Finally, as described in section 5.3.3.4, the use of Lévy mutation operator can use a number of values of α parameter to generate different random numbers by dynamically updating (5.22). This can lead to a large variation in step size and subsequent generation of distinct individuals. This enhances the chance of producing a uniform distribution of Pareto solutions than other MBSO methods (MBSO-G, MBSO-C and MMBSO) provided that the α parameter is properly chosen.

Further, to demonstrate the diversity changes during the search of peer multiobjective optimization algorithms in solving the three design examples, mean HV values are evaluated over a number of function evaluations and the results are shown in Figure 5.15. It can be observed from Figure 5.15 that with the increase in number of function evaluations, the HV values increase and remain consistent as the population converge towards optimal Pareto front. It is

observed that IMBSO performs better than other peer algorithms and the proposed methods while optimizing the three design examples.

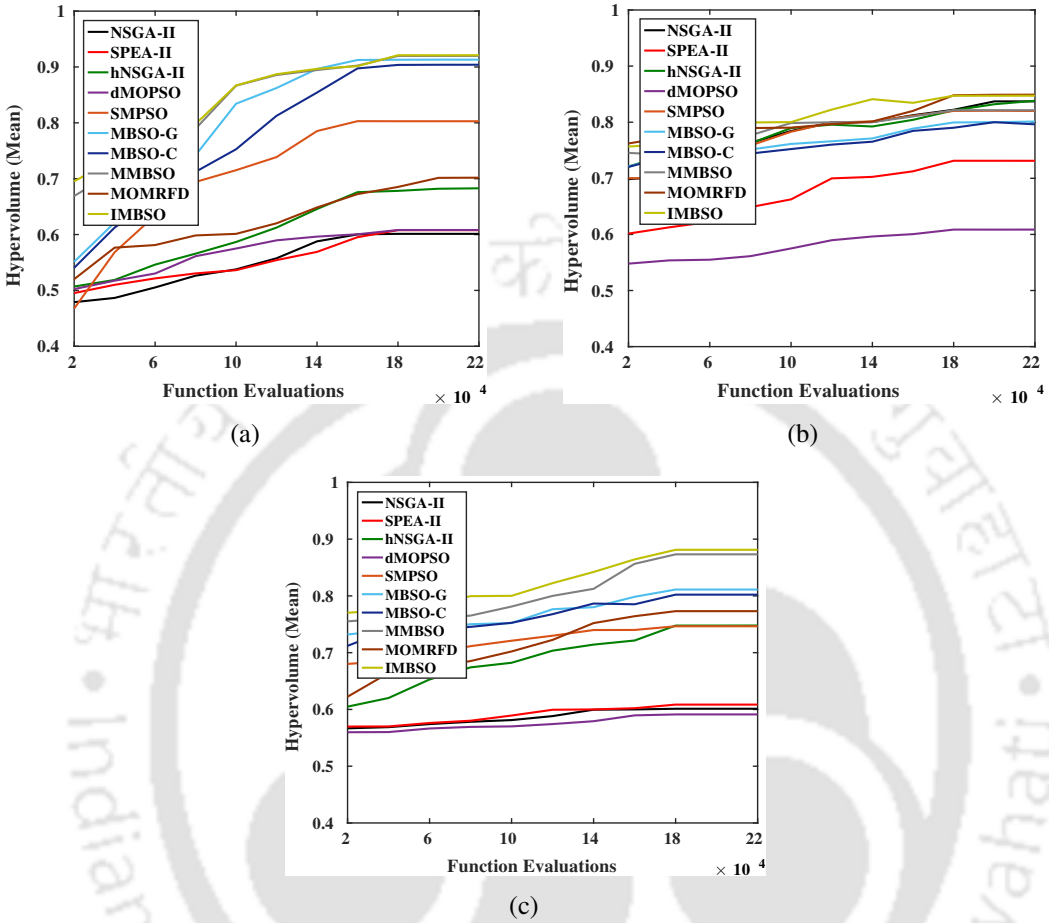


Figure 5.15: Mean HV values with respect to function evaluations of NSGA-II, SPEA-II, *h*NSGA-II, dMOPSO, SMOPSO, MBSO-G, MBSO-C, MMBSO, MOMRFD and IMBSO on (a) two stage operational amplifier, (b) folded cascode amplifier, (c) low noise amplifier.

5.8 Summary

In this chapter, single objective optimization based on both RFD and MRFD algorithms is presented to optimize a CMOS two-stage operational amplifier circuit and a low noise amplifier circuit. Experimental results show that both RFD and MRFD obtain competitive results as compared to other standard optimization techniques. In view of this, RFD based optimization techniques can become one of the preferred choice of algorithms in practice. Still the effectiveness of the scheme is to be seen while solving multiobjective problems.

Therefore, three constrained multiobjective optimization methods are presented in the later part of this chapter to extend the single objective optimization method for optimization of three design examples.

At first, a new hierarchical scheme in polynomial mutation operation is employed in NSGA-II algorithm (hNSGA-II) to propose a new multiobjective optimization algorithm, called as hNSGA-II. At second, single objective optimization framework of MRFD algorithm is extended to develop a multiobjective framework, called as MOMRFD. Finally, a number of improvements are introduced both in clustering (k-means$++$) and population generation (RFD and adaptive L$\acute{e}$vy mutation operator) steps of MBSO algorithm to develop an improved MBSO framework. To showcase the applicability and effectiveness of proposed multiobjective optimization methods and for proof of the concept, three different design examples are optimized. It is observed that continuous Pareto fronts are obtained while optimizing analog/RF circuits using the proposed multiobjective methods. The application of constrained optimization using hNSGA-II, MOMRFD and IMBSO methods is extended to solve standard multiobjective problems to demonstrate the effectiveness of the proposed methods, which is described in Appendix C. It is observed that IMBSO has shown better performance than proposed methods and other peer multiobjective algorithms in terms of convergence and diversity of solutions.

6

IR Drop Minimization in Memristor Crossbar Array

Contents

6.1 Introduction

In this chapter, the motivation is to address the issue of IR drop across memristor crossbar array (MCA) as a single objective minimization problem and to solve the optimization problem using the proposed single objective optimization framework based on both RFD and MRFD methods subject to a number of constraints, such as leakage current, IR drop threshold, metal width, etc. The purpose of using these methods is to efficiently explore the feasible space of operation with an inert flexibility to exploit alternate search spaces for diversity preservation. To showcase the improvement in reduction of IR drop across MCA, experiments have been performed on several MCA benchmark circuits for efficient evaluation of optimal IR drop.

The rest of the chapter is organized as follows. An overview of memristor crossbar array and the analytic formulation is presented in section 6.2 along with IR drop profile of memristor crossbar array. The IR drop minimization problem is described in section 6.3. The performance of RFD and MRFD methods on memristor benchmarks is presented in section 6.4. Section 6.5 summarizes the work done in this chapter.

6.2 Memristor Crossbar Array (MCA)

The structure and characteristics of MCA are presented briefly in this section. Our objective is to study the impact of worst-case voltage variations across MCA and accurately model these variations by using voltage drop statistics along metal wires in MCA. Here, the structure of MCA is presented to be a crossbar arrangement of a set of horizontal and vertical nanowires with memristors located at each cross-section/junction [50] as shown in Figure 6.1(a). For mitigating the amount of leakage current, access transistors (buffers) are placed alongside each memristor as reported in [170]. A resistor modeled network with multiple metal layers (two/three) is considered for IR drop estimation. Each access transistor is modeled as a resistor with arbitrary weights [170] and one of the nodes in the network is connected to the ground through a load resistance. The purpose is to have a selected read/write path initially

to start the process of optimization. All memristors are modeled as resistors with effective memristance values as reported in [170]. For realistic approach, the static resistance of the

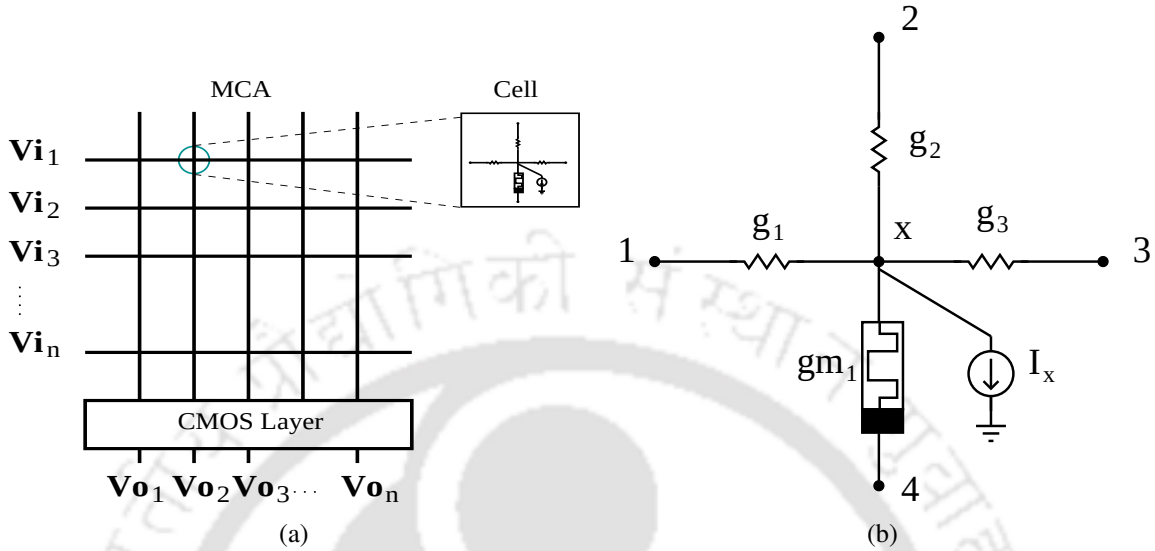


Figure 6.1: (a) Memristor crossbar array architecture [50], (b) Resistive model of a cell in MCA.

memristor is considered to be the total charge flowing throw the memory element. The metal wires along crossbar array are assumed to have nominal resistances to present the occurrence of IR drop in MCA. In this regard, each wire segment in MCA can be modeled as a series connection of resistors and memristors connected to other nodes (or ground). In each cell of MCA as shown in Figure 6.1(b), by applying Kirchoff's current law (KCL) at center node x,

$$V_x = \frac{\sum_{i=1}^{deg(x)-1} g_i V_i}{\sum_{i=1}^{deg(x)-1} g_i + gm_1} + \frac{gm_1 V_4}{\sum_{i=1}^{deg(x)-1} g_i + gm_1} - \frac{I_x}{\sum_{i=1}^{deg(x)} g_i + gm_1}, \tag{6.1}$$

where V_x and I_x represent the voltage and current drawn at node x, respectively; g and gm_1 denote the conductances of resistor and memristor, respectively, and $deg(x)$ is the number of neighbors of node x. In view of this, total current across MCA can be expressed as (considering steady state analysis) [171],

$$I = [\mathcal{S} \;\; \mathcal{G}]\, \mathcal{V} + \hat{I}_{\mathcal{L}},$$
$$I, \mathcal{V}, \hat{I}_{\mathcal{L}} \in \mathbb{R}_{m\times 1},\; \mathcal{S}, \mathcal{G} \in \mathbb{R}_{m\times m} \tag{6.2}$$

where $\mathcal{S}$ represents memristor conductance matrix (considering doping ratio); $\mathcal{G}$ denotes wire conductance matrix; $\mathcal{V}$ is a vector representing port voltages; $\hat{I}_{\mathcal{L}}$ $(= \sum_{x=1}^{N} I_x)$ is the vector of

current sources connected to ground (leakage current) having N nodes; I denotes vector of current flowing through different ports, and m is the number of ports (horizontal or vertical crossbars).

Theorem 5. *The system of equations, $[\mathcal{S}\ \mathcal{G}]\mathcal{V}=I-\hat{I}_{\mathcal{L}}$ (described in (6.2)) governed by Modified Nodal Analysis (MNA) to analyze MCA network, can be expressed as,*

$$\begin{bmatrix} A_{r\mathcal{S}} & A_{r\mathcal{G}}GA_{r\mathcal{G}}^T & A_{r\mathcal{T}} \\ 0 & \beta A_{r\mathcal{T}}^T & \alpha \end{bmatrix} \begin{bmatrix} i_{\mathcal{S}} \\ v_n \\ i_{\mathcal{T}} \end{bmatrix} = \begin{bmatrix} -A_{\mathcal{L}}\hat{i}_{\mathcal{L}} \\ S_T \end{bmatrix} \tag{6.3}$$

where $A_{r\mathcal{S}}$, $A_{r\mathcal{G}}$ and $A_{r\mathcal{T}}$ represent reduced incidence matrices across memristor, conductance and independent voltage sources respectively, and $A_{\mathcal{L}}$ denotes the incidence matrix of leakage current.

(i) As the MCA network is divided into current source type branches (J), conductance type branches ($\mathcal{G}$), branches having memristors ($\mathcal{S}$) and the remaining branches ($\mathcal{T}$), in the first step Kirchoff's Current equations (KCE) are formulated for the network [106].

$$\begin{aligned} A_r i &= 0 \\ \begin{bmatrix} A_{r\mathcal{S}} & A_{r\mathcal{G}} & A_{r\mathcal{T}} \end{bmatrix} \begin{bmatrix} i_{\mathcal{S}} \\ i_{\mathcal{G}} \\ i_{\mathcal{T}} \end{bmatrix} &= -A_{rJ} i_J \ \ (KCE) \end{aligned} \tag{6.4}$$

(ii) In second step, the device characteristic equations are written for each device of the network.

$$\begin{aligned} \text{Let } i_{\mathcal{G}} &= G v_{\mathcal{G}} \ \text{(Device characteristics)} \\ i_{\mathcal{S}} &= G_{\mathcal{S}} v_{\mathcal{S}} \ \text{(Ohm's law for memristor [172])} \end{aligned} \tag{6.5}$$

Then, (6.4) can be rewritten as,

$$\begin{bmatrix} A_{r\mathcal{S}}G_{\mathcal{S}} & A_{r\mathcal{G}}G & A_{r\mathcal{T}} \end{bmatrix} \begin{bmatrix} v_{\mathcal{S}} \\ v_{\mathcal{G}} \\ i_{\mathcal{T}} \end{bmatrix} = -A_{rJ} i_J \tag{6.6}$$

(iii) Finally, the branch voltages are replaced by their corresponding node potentials in the device characteristic equations (Kirchoff's voltage equation (KVE)).

$$\begin{bmatrix} v_{\mathcal{S}} \\ \dots \\ v_{\mathcal{G}} \\ \dots \\ v_{\mathcal{T}} \\ \dots \\ v_J \end{bmatrix} = \begin{bmatrix} A_{r\mathcal{S}}^T \\ \dots \\ A_{r\mathcal{G}}^T \\ \dots \\ A_{r\mathcal{T}}^T \\ \dots \\ A_{rJ}^T \end{bmatrix} v_n \ \ (KVE) \tag{6.7}$$

Considering (6.7), (6.6) can be expressed as,

$$\begin{bmatrix} A_{r\mathcal{S}} G_{\mathcal{S}} A_{r\mathcal{S}}^T & A_{r\mathcal{G}} G A_{r\mathcal{G}}^T & A_{r\mathcal{T}} \end{bmatrix} \begin{bmatrix} v_n \\ v_n \\ i_{\mathcal{T}} \end{bmatrix} = -A_{rJ} i_J \tag{6.8}$$

The device characteristic of the branches in $\mathcal{T}$ can be expressed as [106],

$$\begin{bmatrix} \alpha & \beta \end{bmatrix} \begin{bmatrix} v_{\mathcal{T}} \\ i_{\mathcal{T}} \end{bmatrix} = S_{\mathcal{T}} \tag{6.9}$$

Further, using (6.9), (6.8) can be rewritten as,

$$\begin{bmatrix} A_{r\mathcal{S}} G_{\mathcal{S}} A_{r\mathcal{S}}^T & A_{r\mathcal{G}} G A_{r\mathcal{G}}^T & A_{r\mathcal{T}} \\ 0 & \beta A_{r\mathcal{T}}^T & \alpha \end{bmatrix} \begin{bmatrix} v_n \\ v_n \\ i_{\mathcal{T}} \end{bmatrix} = \begin{bmatrix} -A_{rJ} i_J \\ S_{\mathcal{T}} \end{bmatrix} \tag{6.10}$$

Equation (6.10) represents the analytic formulation of MCA network using MNA for the linear static case, where $(A_{r\mathcal{S}} G_{\mathcal{S}} A_{r\mathcal{S}}^T) v_n = A_{r\mathcal{S}} i_{\mathcal{S}}$.

In this regard, if **v** is the vector of voltage drops (IR drop) at each node of MCA network,

then considering (6.3), $\mathbf{v}$ can be expressed as (considering $A_{rJ}i_J = A_{\mathcal{L}}\hat{i}_{\mathcal{L}}$),

$$\begin{aligned}\mathbf{v} &= (V_{dd} - v_n) \\ &= V_{dd} - (A_{r\mathcal{G}}GA_{r\mathcal{G}}^T)^{-1}(A_{\mathcal{L}}\hat{i}_{\mathcal{L}} + A_{r\mathcal{T}}i_{\mathcal{T}} + A_{r\mathcal{S}}i_{\mathcal{S}})\end{aligned} \tag{6.11}$$

6.2.1 IR drop profile

For obtaining an optimal $\mathbf{v}$ across MCA, it is necessary to have a precise estimation of node potentials (v_n) across the crossbar by generating a system matrix using MNA and by analyzing it using LU decomposition method. For the sparse LU decomposition of system matrix, SUPERLU [173] and KLU [132] solvers may be utilized.

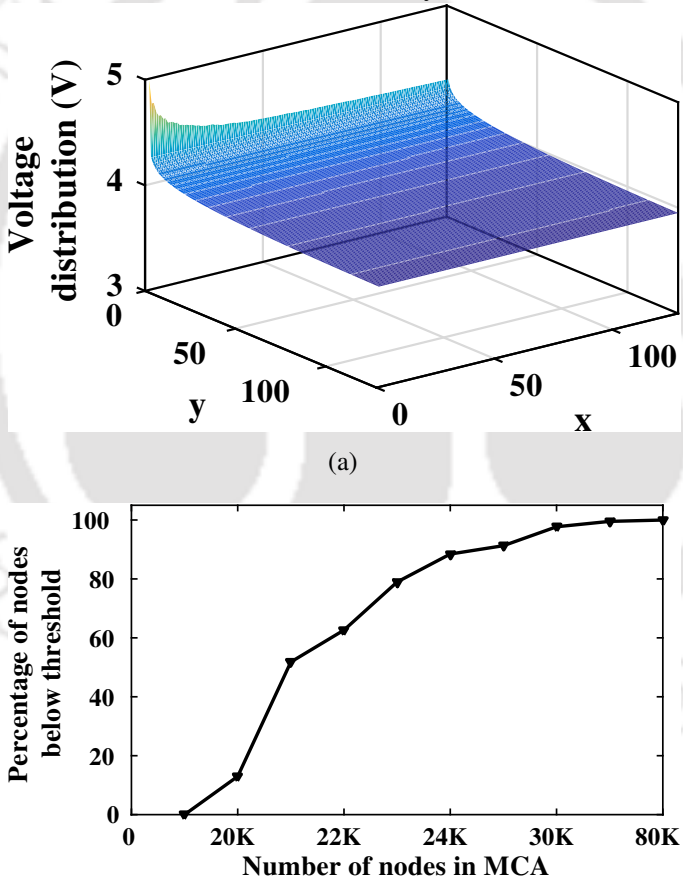


Figure 6.2: (a) Voltage distribution across an MCA of size 128×128, (b) Percentage of nodes below threshold for different MCAs.

Here, KLU solver is employed to solve the system matrix in (6.3) for efficient estimation of node potentials (v_n). Once the node potentials are obtained, the corresponding voltage

drops at each node are evaluated using (6.11). To analyze the occurrence of IR drop (voltage drop) in MCA network, different sizes of MCAs are employed for our implementation. A general crossbar model is utilized for the experiments as shown in Figure 6.1(a). For this model architecture, voltage reaching the far end of the crossbar array may be considerably lower than the supply voltage for a crossbar of size beyond 64×64 [11]. Since the amount of IR drop depends on the position of supply units and resistance states of memristors, the largest IR drop occurs at the far end of the crossbar (see Figure 6.2(a)). However, IR drops across other nodes of the crossbar below certain threshold are also determined during the process of optimization for the accurate estimation of reliability. The amount of threshold is estimated through standard probability distribution statistics of IR drop across the crossbar, with bounds being determined by the worst and the best case occurrence of IR drop ($\pm 10\%$ of power supply voltage (V_{dd})). Figure 6.2(b) shows the percentage of nodes below a certain threshold with increase in the size of MCAs. It can be observed from Figure 6.2(b) that with increase in the size of MCA networks, more number of nodes are prone to worst-case voltage drops (IR drops) and this may affect functionality of the underlying logic blocks significantly.

6.3 Problem Formulation

In this section, supply voltage variations across MCA is modeled as a minimization problem to accurately map IR drop statistics across MCA. It is assumed that the topology and physical locations of MCA are fixed. For minimizing the number of sneak paths across the crossbar array, unselected lines are grounded instead of being keeping floated. However, the effect of sneak path current ($\hat{I}_{\mathcal{L}}$) cannot be fully ignored due to the voltage drops (IR drop) across nanowires [170]. Therefore, it is crucial to adjust reference current (I_{ref}) based on the occurrence of sneak paths across MCA. Both metal width of nanowires and wire resistance have statistical measures of the changes in the voltage drop profile across MCA. Long nanowires (both selected and unselected) make signal path critical by introducing voltage drop and to minimize IR drop, metal area is subjected to constraints, i.e. by adjusting metal

width or length accordingly during objective function formulation of IR drop minimization. Further, IR drop at any node of MCA network is bounded by a voltage drop constraint, threshold of IR drop (V_{th}). Here, V_{th} is evaluated by determining the average IR drop occurrence of entire MCA network. To capture the impact of IR drop on the performance of MCA, a posynomial function f defined in space $\mathcal{C}$ can be represented as,

$$f_{\mathbf{v}} = \underset{\mathbf{v} \in \mathcal{C}}{minimize}\ \mathbf{v} = \underset{\{\gamma_x, \delta_x, v_x\} \in \mathcal{C}}{minimize}\ \sum_{x=1}^{N} \vartheta_x, \tag{6.12}$$

where N is the number of nodes in MCA, ϑ_x is a non-negative integer which spans $\mathbf{v}$ in the finite-dimensional subspace, such that each IR drop element $\vartheta_x \in \mathcal{C}$ is linearly independent. In this regard, the optimal voltage drop at different nodes of MCA can be estimated through the minimization problem defined in (6.12) subject to the constraints shown in (6.13),

$$\text{subject to} \begin{cases} 0 \leq A_{\mathcal{L}} \hat{i}_{\mathcal{L}} \leq I_{ref},\ 0 \leq A_{\mathcal{S}} i_{\mathcal{S}} \leq I_m, \\ 50\Omega \leq R^m_{on} \leq R^{on}_{ref},\ 100\Omega \leq R^m_{off} \leq R^{off}_{ref}, \\ 10nm \leq d_m \leq D^m_{ref}, \\ M_{area} \leq \sum_{(x,y)\in R}\ \rho l^2_{xy} g_{xy},\ \rho l_{xy} g_{xy} \geq w_{min}, \\ V_{dd} - v_n \leq \vartheta_{th},\ \ \forall n \in N, \end{cases} \tag{6.13}$$

where I_{ref} is the total current along a crossbar row or column; I_m is the maximum allowable current across a memristor; R^m_{on} is the resistance of memristor in the conducting state; R^m_{off} denotes resistance of the memristor in the non-conducting state; d_m is the thickness of memristor film; V_{th} denotes the threshold of IR drop; ρ and w_{min} denote the sheet resistance and minimum metal width of any branch, respectively; l_{xy} and w_{xy} are length and width of (x, y) branch of MCA, respectively.

6.4 Minimization of IR drop of Memristor Crossbar Array

6.4.1 Memristor Crossbar Array Benchmarks

Due to unavailability of industry benchmarks, we have developed our own benchmark circuits for testing our proposed methodology. We have followed the notion of having minimum interconnection complexity without violating Rent's rule [174] while developing the benchmark circuits. For showcasing IR drop issue prominently in MCAs, the requirement is to analyze large benchmark circuits. This allows the use of optimization methods, i.e., RFD and MRFD to be tested efficiently with respect to the characteristic parameters of MCAs if the underlying network is connected and regular. The memristors are represented by interconnected blocks (representing dynamic resistance states) and the entire network is connected through a series of multi-terminal nets. Although the graph of entire MCA network is a randomly connected circuit, the nodes are numbered in a structured manner to minimize irregularity. The regularity in the benchmark circuits is maintained by limiting the interconnection complexity without further sub-dividing nets into sub-nets. The memristor blocks are placed between horizontal and vertical bars to equivalently map a neuromorphic environment. In this regard, these benchmarks may be used to analyze neuromorphic computing system with the addition of realistic logic blocks. The details on MCA benchmarks are listed in Table 6.1.

6.4.2 Preprocessing Stage

Before optimization, steady-state analysis is performed to evaluate potential at each node (or cross-section) of MCA using KLU solver and the maximum, minimum and average values of IR drop are evaluated. The IR drop values (Δv_{ir}) are reported for each MCA benchmark circuit in Table 6.1 along with the computational cost involved in analyzing MCA benchmarks using KLU solver. Once steady state analysis is performed for the entire MCA, it is subjected to optimization process. Due to huge size of MCA circuit, solving it directly for optimization is extremely difficult and time-consuming. Therefore, we propose to reduce the problem size by choosing a suitable voltage threshold (V_{th}), i.e., performing analysis for nodes whose

Table 6.1: Statistics of different MCA benchmarks before optimization and computational time required for steady state analysis using KLU solver.

Crossbar benchmarks	Benchmark Statistics[1]				Δv_{ir} (V)			ϑ $(\%)$[2]	KLU Time (sec)
	$\#n$	$\#r$	$\#m$	$\#s$	Max.	Min.	Avg.		
IITmcag1	5K	4.8K	2.5K	4.49K	0.2914	0.0727	0.2721	0.0	0.001
IITmcag2	20K	19.2K	10K	16.43K	0.6487	0.1460	0.6123	12.98	0.005
IITmcag3	25K	23.58K	12.56K	21.25K	0.7369	0.1633	0.6973	61.27	0.005
IITmcag4	30K	28.82K	14.65K	25.32K	0.8237	0.1798	0.7787	97.70	0.006
IITmcag5	40K	37.84K	21.02K	34.20K	0.9477	0.2032	0.8989	99.52	0.006
IITmcag6	50K	48.30K	26.12K	42.53K	1.0781	0.2276	1.0238	99.8813	0.007
IITmcag7	80K	76.78K	40.10K	74.35K	1.4293	0.2925	1.3495	99.9838	0.008
IITmcag8	100K	95.56K	50.12K	85.65K	1.5910	0.3210	1.5155	99.9890	0.010
IITmcag9	180K	170.56K	92.53K	144.56K	2.2598	0.4391	2.1566	99.9978	0.030
IITmcag10	320K	307.23K	158.34K	271.21K	3.1208	0.5859	2.9861	99.9991	0.070
IITmcag11	500K	478.07K	254.23K	425.50K	3.2975	0.6309	3.2200	99.9994	0.100
IITmcag12	1M	964.01K	483.02K	899.60K	3.3000	0.6332	3.2623	99.9997	0.170

[1] $\#n$ denotes number of nodes, $\#r$ denotes number of resistances, $\#r$ denotes number of memristors, $\#s$ denotes number of current sources and supply voltage sources.
[2] ϑ denotes the percentage of nodes below threshold (V_{th}) before optimization.

potentials are below a certain threshold. This reduced MCA circuit is analyzed directly for the minimum IR drop by evaluating potentials at each node subjected to design constraints. Since the occurrence of the maximum IR drop is anticipated at the far end of MCA, corresponding nodes are retained for analysis during optimization.

6.4.3 Peer Algorithms and Parameter Settings

With aim of demonstrating the applicability and effectiveness of ALC-PSO algorithm, four different state-of-art optimization algorithms, $i.e.$, gGA [144], DE [59], PSO [67], CRPSO [73, 74], Lévy-PSO [76, 77], ALC-PSO [78] are employed for evaluation of the minimum IR drop across different MCA benchmarks. Authors' implementations of all the algorithms are used during evaluations and comparison of solutions. As Levy-PSO can avoid premature convergence by incorporating long jumps during particle movement, it is selected for the evaluations and comparison. With the incorporation of aging and challenging mechanisms to select a leader, ALC-PSO differs from the conventional PSO algorithm and it is proved to be efficient in solving both unimodal and multimodal functions without sacrificing convergence

speed and reliability [78]. Therefore, we choose to employ ALC-PSO algorithm as one of the algorithms for comparison. The details on different control parameters and corresponding values of these algorithms and proposed RFD and MRFD algorithms are listed in Table 6.2.

6.4.4 Results and Discussion

In this section, we provide simulation results of RFD and MRFD methods and other peer algorithms after the minimization of IR drop in different MCA benchmarks. Further, we discuss and compare the performance of the proposed methods with other standard optimization algorithms. Both RFD and MRFD algorithms are implemented by using C/C++ language and all simulation experiments are performed using Linux environment on a computer having Intel Xeon E5-2620 processor with 64GB of RAM. Since the problem formulation of IR drop problem in MCAs is similar to that of IR drop problem in power distribution networks, the optimization framework having RFD and MRFD methods as described in chapter 4 has been opted during implementation without making any modifications to the algorithmic structure.

Due to the large size of MCA network, it is essential to keep track of the crossbar area (M_{area}) during optimization. In the optimization engine, metal conductance (g), memristor conductance ($gm = 1/R_{on}$ or $1/R_{off}$) and the load current (i_L) are considered as design variables (χ) by imposing specific bounds as described in section 6.3. Once the variables change during optimization, metal width (w_m) also varies accordingly forcing the crossbar area to constrain itself within specific limits. Since general implementations of memristors are limited to emulation in theory, our implementation assumes a simple model for memristor

Table 6.2: Control parameters of gGA, DE, PSO, CRPSO, Lévy-PSO, ALC-PSO, RFD and MRFD algorithms.

Parameters	gGA	DE	PSO	CRPSO	Lévy-PSO	ALC-PSO	RFD	MRFD
Population size (P)	100	100	100	100	40	20	100	100
Max. function evaluation (FE_{max})	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05
Mutation, mutation rate	Gaussian, 0.05	$DE/current\text{-}to\text{-}best/1$, -	-	-	-	-	-	-
Crossover, crossover probability	SBX, 0.9	$rand/1/exp$, 0.9	-	-	-	-	-	-
Selection, selection probability	Roulette, 0.33	-	-	-	-	-	-	-
c_1, c_2, v_{max}, v_{min}	-	-	1.4, 1.4, -5, 5	2.05, 2.05, 0.01, 1.0	2.0, 2.0, -, -	2.0, 2.0, 0.01, -	-	-
w^{max}, w^{min}/w	-	-	1.0, 0.4/-	-	-, -/$\frac{(Max\_iter-iter)}{iter}$	-, -/0.4	-	-,-/(1.0,2.0)
P_{cr}, $v_{craziness}$	-	-	-	0.3, 0.0001	-	-	-	-
C_r, F	-	0.9, 0.8	-	-	-	-	-	-
Θ_0, T, pro	-	-	-	-	-	60, 2, $1/n$	-	-
β	-	-	-	-	rand(0,2)	-	-	-
$Erosion\ parameter$	-	-	-	-	-	-		

realization inside the crossbar as described in section 6.2. The number of inputs (supply voltage) is limited to one or two and all bias voltages are kept within limits without changing the characteristics of the crossbar. The input specifications are listed in Table 6.3 for reference.

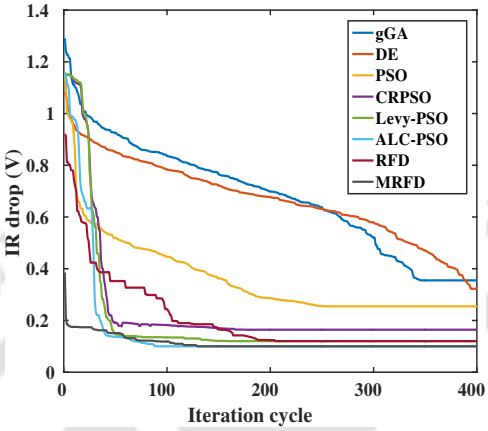


Figure 6.3: Comparison of evolution curves of IR drop values for a single cell of IITmcag12 benchmark.

All specifications are considered to have a realistic environment. As the optimization is carried out on each cell (whose potential is below threshold) of MCA benchmarks, it is exhaustive to demonstrate the results of each cell and corresponding specifications. However, we bound the specifications within specific limits to reduce the IR drop for all benchmarks as listed in Table 6.4. For mitigating the occurrence of any local convergence, both boundary and functional constraints are handled carefully during simulations using random constraint method [115] and constrained-domination principle [7], respectively. The details on these two constraint handling methods are described in section 5.6.

Table 6.3: Input specifications

Input Parameters	Value/Constraints/Bound
No. of decision variables (ϕ)	$3-4$
Wire conductance (g)	$[0.05, 0.5]$ $\mho$
Memristor conductance ($gm = (1/R_{on})$ or $(1/R_{off})$)	$[0.05, 0.5]$ or $[0.01, 2.0]$ $\mho$
Supply voltage (V)	$5V$
Threshold voltage (V_{th})	$\leq 3.5V$
Min. width (w_{min})	$\leq 50nm$
Total current I_{ref}	$\leq 1mA$
Max. current across memristor I_m	$\geq 1\mu A$
Maximum FEs/single cell optimization (FEs)	1e+05
No. of independent runs (r_{ind})	25

Table 6.4: Values of different decision variables and specifications of a single cell in `IITmcag12` MCA benchmark after optimization.

Decision variables/ Specifications	Values / constraints/ bound	gGA	DE	PSO	CRPSO	Lévy-PSO	ALC-PSO	RFD	MRFD
ϑ_x (V)	≤ 0.5	0.32	0.36	0.25	0.16	0.12	0.10	0.12	0.10
g_1 $(\mho)$	$[0.05, 0.5]$	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
g_2 $(\mho)$	$[0.05, 0.5]$	0.45	0.46	0.4178	0.38	0.371	0.3435	0.352	0.341
g_3 $(\mho)$	$[0.05, 0.5]$	0.0487	0.0471	0.0434	0.0368	0.035	0.0351	0.035	0.0352
gm_1 $(\mho)$	$[0.05, 0.5]$	0.4995	0.4991	0.4991	0.49	0.49	0.49	0.49	0.49
I_{ref} (mA)	≤ 1	0.96	0.95	0.96	0.98	1	1	1	1
I_m (μA)	≥ 1	10.2	9.46	11.45	12.0	12.0	12.0	12.0	12.0
w_{min} (nm) (Four branches)	≤ 50	50.07, 45.31, 4.87, 49.98	50.07, 46.03, 4.71, 49.94	50.07, 41.8, 4.34, 49.94	50.07, 38.02, 3.68, 49.03	50.07, 35.13, 3.6, 49.03	50.07, 34.32, 3.51, 49.03	50.07, 34.70, 3.51, 49.03	50.07, 34.44, 3.51, 49.03
V_{th} (V)	≤ 3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5
Iterations	$\leq$ 1e+05	400	350	256	180	163	162	220	160
Time (sec)	-	1.2	1.1	0.4	0.2	0.17	0.17	0.38	0.17

The results of input specifications for a single cell of IITmcag12 benchmark obtained after optimization using RFD, MRFD and other peer algorithms are also listed in Table 6.4. Further, to demonstrate the applicability of both the proposed methods, the evolution curves of IR drop values for a single cell of IITmcag12 benchmark are shown in Figure 6.3 for all peer algorithms. It can be observed from Figure 6.3 that both RFD and MRFD successfully minimize the IR drop of a single cell of IITmcag12 benchmark. However, it can also be observed that MRFD, ALC-PSO, Lévy-PSO and CRPSO show promising improvement in demonstrating rapid convergence over reasonable amount of time. At the beginning of search process, the evolution curve of MRFD is similar to RFD, ALC-PSO, Lévy-PSO, CRPSO and PSO. However, as additional factors are employed in MRFD, it shows a promising pattern in search movement, where it quickly jumps to the solution (observed as a sharp drop in the evolution curve in Figure 6.3). This represents the capability of MRFD to obtain a near-optimal solution while minimizing the IR drop of a single cell of MCA network over a reasonable amount of time.

For demonstrating the applicability of MRFD in minimizing IR drop of entire MCA network, each cell of MCA benchmark is subjected to optimization using MRFD method. The results obtained after optimization of cells of different MCA benchmarks are listed in Table 6.5. The comparison of solution accuracy on MCA benchmarks is performed by evaluating the percentage of number of cells having potentials below threshold (ϑ') and the percentage of reduction in the area of entire network ($A = M_{area}$) after optimization. It can be ob-

Table 6.5: Statistics of Optimal IR drop for different MCA benchmarks.

Crossbar Benchmarks	gGA		DE		PSO		CRPSO		Lévy-PSO		ALC-PSO		RFD		MRFD	
	$\vartheta'(\%)^1$	$A(\%)^1$	$\vartheta'(\%)^1$	$A(\%)^1$	$\vartheta'(\%)^1$	$A(\%)^1$	$\vartheta'(\%)^1$	$A(\%)^1$	$\vartheta'(\%)^1$	$A(\%)^1$	$\vartheta'(\%)^1$	$A(\%)^1$	$\vartheta'(\%)^1$	$A(\%)^1$	$\vartheta'(\%)^1$	$A(\%)^1$
IITmcag1	0.0011	0.81	0.0011	0.81	0.0011	0.81	0.0011	0.81	0.0011	0.81	0.0011	0.81	0.0011	0.81	0.0011	0.81
IITmcag2	0.0549	2.87	0.0549	2.87	0.0549	2.87	0.0120	3.03	0.0120	3.03	0.0120	3.03	0.0549	2.87	0.0120	3.03
IITmcag3	3.6154	1.35	3.1151	1.67	1.2961	2.04	1.0231	2.44	0.9121	2.68	1.1610	2.20	1.2610	2.11	1.1310	2.18
IITmcag4	3.6667	1.67	3.6980	1.05	1.7886	2.56	1.4224	3.15	0.9401	4.30	0.9617	4.05	1.9034	2.00	0.9617	4.05
IITmcag5	5.2096	2.02	3.9325	2.38	1.3884	3.87	0.9871	5.07	0.7310	5.25	0.6695	5.56	1.1695	4.80	0.6521	5.60
IITmcag6	7.7900	1.34	5.9700	1.67	1.9761	4.00	1.0912	4.77	1.0010	4.71	1.7226	4.33	1.2510	4.59	1.0010	4.71
IITmcag7	5.8537	2.23	4.0587	2.56	2.5870	3.56	1.3871	5.14	1.2212	5.28	1.2212	5.28	1.3871	5.14	1.2010	5.30
IITmcag8	7.8520	1.42	6.1224	1.77	2.0506	4.33	1.2091	6.00	1.1250	6.21	1.0507	6.44	1.8120	4.80	1.0507	6.44
IITmcag9	7.8933	1.81	7.0933	1.02	2.3501	4.13	1.1210	5.34	1.0102	5.89	0.9722	6.02	1.8691	4.82	0.9012	6.10
IITmcag10	9.4203	2.90	7.9446	3.03	3.0862	4.67	1.8712	6.12	1.7491	6.33	1.7491	6.33	2.2160	5.00	1.7491	6.33
IITmcag11	9.5910	3.66	7.5620	4.61	3.2215	6.22	2.6667	7.67	2.2391	7.80	2.0480	8.00	3.0101	6.80	1.9211	8.34
IITmcag12	12.1524	5.67	11.3200	7.87	5.8126	9.34	4.2866	10.45	4.0121	10.66	3.8480	11.01	5.2091	9.60	3.6703	11.41

[1] ϑ' denotes percentage of affected nodes (above threshold) after optimization. 'A' denotes percentage of reduction in wire area after optimization.

[2] 'Time' denotes computational time (algorithm time + process time) required for entire optimization process.

served from Table 6.5 that MRFD performs significantly better than other peer algorithms on MCA benchmarks except IITmcag5, IITmcag6 and IITmcag8 benchmarks. However, it is observed that CRPSO performs better on IITmcag6 benchmark and ALC-PSO performs better on IITmcag5 and IITmcag8 benchmarks as compared to other peer algorithms in minimizing IR drop of these benchmarks.

Moreover, to demonstrate the reliability and search speed of MRFD and other peer algorithms in solving MCA benchmarks, the number of function evaluations (FEs) and the computational time required to obtain acceptable solutions of all cells of MCA benchmarks are reported in Table 6.6. The reliability of search is also reflected by the percentage of successful runs ("S% ") of an algorithm in obtaining acceptable solutions. The ranks of all the algorithms in solving MCA benchmarks are evaluated by the descending order of "S%" and by ascending order of FEs. It can be observed from Table 6.6 that the number of FEs and computational time (T) required to achieve acceptable solutions of MCA benchmarks are significantly different for all the peer algorithms, e.g., MRFD requires less than $4e$+09 FEs and around 390 minutes to optimize IITmcag12 benchmark, whereas RFD, ALC-PSO, Lévy-PSO, CRPSO, PSO, DE and gGA require more than $4e$+09 FEs to yield acceptable results within 400 minutes. Further, it can be observed from Table 6.6 that MRFD has shown competitive performance with respect to ALC-PSO, Lévy-PSO and CRPSO algorithms and superior performance than other peer algorithms.

Further, all peer algorithms have shown a poor success rate as compared to MRFD (except ALC-PSO, which has shown similar success rate with MRFD). One of the reasons for this poor success rate may be due to poor parameter calibration inside the algorithm and the degree of randomness involved in these parameters. With increase in randomness and size of search space, the uncertainty in obtaining near optimal solutions is quite high and it also affects the overall success rate of a metaheuristic (as shown in Table 6.6). On the other hand, MRFD offers 100% success rate on nine of the twelve MCA benchmarks except for IITmcag10, IITmcag11 and IITmcag12 benchmarks. Since the size of these three MCA benchmarks (IITmcag10, IITmcag11, IITmcag12) are quite large as compared to the other MCA benchmarks, it is difficult to handle a large number of local minima. It is not necessary for algorithms, like MRFD to obtain an acceptable solution steadily in all independent attempts. However, on the prospect of search speed, MRFD can be observed as one of the fastest algorithms in terms of number of FEs and computational cost. Further, a composite ranking is evaluated to test the performance of all algorithms on MCA benchmarks and it can be observed that MRFD has an average rank of 1.25, which is the smallest among all. This demonstrates that MRFD shows competitive performance over other peer algorithms in minimizing IR drop of different MCA benchmarks.

6.5 Summary

In this chapter, IR drop problem in memristor crossbar array is addressed with an aim to reduce reliability issues. The problem of IR drop is formulated as a constrained nonlinear single objective problem, which is successfully solved using RFD and MRFD methods. Decision variables are updated iteratively in both RFD and MRFD to minimize IR drop by reducing the resistance discrepancy and to have a uniform programming voltage distribution (reduced IR drop) across MCA. For a pre-designed crossbar array, it is necessary to analyze reliability of the crossbar design by the knowledge of IR drop across different locations of the crossbar array. This enables the need of efficient optimization methods, like RFD, MRFD,

Table 6.6: Search speed and reliability comparisons on MCA benchmarks.

Crossbar Benchmarks	gGA				DE			PSO			CRPSO			Lévy-PSO			ALC-PSO			RFD			MRFD		
	S%[1]	FEs[1]	T[1]	R[1]	S%	FEs	T	R	S%	FEs	T	R	S%	FEs	T	R	S%	FEs	T	R	S%	FEs	T	R	S%	FEs	T	R	S%	FEs	T	R
IITmcag1	100	12031	63.47s	7	84	13345	68.12s	8	100	10561	40.16s	6	100	7312	28.43s	4	100	7200	25.12s	1	100	7207	26.34s	3	100	8911	35.12s	5	100	7200	25.14s	1
IITmcag2	100	43002	283.23s	8	100	38451	262.56s	7	100	35181	249.08s	6	100	20199	134.51s	4	100	19800	126.04s	2	100	19885	128.78s	3	100	27015	203.06s	5	100	19102	110.80s	1
IITmcag3	100	67153	359.47s	8	100	65434	334.58s	7	100	62801	316.24s	6	100	31211	151.23s	1	100	31289	151.96s	3	100	31519	153.39s	4	100	59016	298.01s	5	100	31211	151.19s	1
IITmcag4	80	81231	427.64s	8	100	81082	423.51s	7	100	77651	379.05s	6	100	40618	182.34s	4	100	39700	172.01s	1	100	39791	174.11s	3	100	60981	342.13s	5	100	39700	172.01s	1
IITmcag5	76	87388	498.93s	8	88	87312	481.45s	7	100	80121	430.36s	5	100	67821	281.01s	4	96	60251	268.06s	3	100	58712	245.87s	1	100	80770	440.87s	6	100	58800	252.07s	2
IITmcag6	84	97612	706.04s	8	88	97199	677.32s	7	96	95109	640.47s	6	100	79023	419.87s	1	96	79520	422.30s	3	100	79612	427.01s	4	96	91200	603.00s	5	100	79214	422.01s	2
IITmcag7	80	127611	22.64m	8	84	110354	19.56m	7	88	100045	17.78m	6	96	90166	13.50m	4	96	89978	12.00m	3	100	89821	12.34m	2	96	95201	16.14m	5	100	89041	12.00m	1
IITmcag8	80	276889	30.19m	8	84	243118	27.89m	7	84	206123	24.01m	5	100	106712	18.23m	4	92	97260	17.12m	3	100	96533	16.88m	1	92	206300	26.88m	6	100	96550	16.80m	2
IITmcag9	68	578321	68.52m	8	80	498127	58.98m	7	84	382311	49.80m	6	96	231987	41.83m	4	96	208021	38.01m	3	100	208191	37.77m	2	92	380100	47.00m	5	100	206100	35.10m	1
IITmcag10	76	876653	119.30m	8	80	748541	99.23m	7	84	598232	86.11m	6	92	498910	76.01m	4	92	468401	71.30s	3	96	462319	69.30m	2	92	560110	82.30m	5	96	461001	68.00m	1
IITmcag11	76	2318761	222.16m	8	80	1102663	180.21m	7	80	982167	151.08m	6	96	881205	122.03m	4	96	832671	118.56m	3	96	829101	113.21m	2	88	970051	140.20m	5	96	820132	110.11m	1
IITmcag12	68	6712154	663.15m	8	76	5311781	543.45m	7	80	4712198	457.60m	6	88	4240911	416.18m	4	92	4182549	408.62m	3	96	4009129	402.09m	2	88	4609010	440.00m	5	96	3998110	390.00m	1
Avg.rank	7.91				7.08			5.83			3.50			2.58			2.41			5.16			1.25								

[1] 'S%' denotes percentage of successful runs out of 25 independent runs, 'FEs' denotes approx. number of function evaluations $\times 100$ (Averaged over successful runs), 'T' denotes computational time required for entire optimization process, 'R' denotes rank. The time units are abbreviated as "s" and "m", which denote "seconds" and "minutes", respectively.

etc. in the design verification MCA networks.

Here, the applicability of the proposed methods is analyzed by solving different MCA benchmarks and, the search speeds and reliability are compared with other peer algorithms to evaluate their effectiveness. It is observed that MRFD has shown a better performance than RFD and other peer algorithms. Although MRFD suffers in obtaining acceptable solutions for large MCA benchmarks with 100% success rate, the performance can be improved by incorporating different diversity preserving schemes [175, 176]. These preservation scehemes can enhance the search capability by efficiently exploring the multidimensional search space. Further, the computational cost in solving an entire MCA network using MRFD can be minimized by employing efficient parallelizable strategies [177] to analyze cells in multicore and manycore environments.

7

Conclusion and Future Work

Contents

7.1 Conclusion

As many real-life optimization problems are difficult to solve by exact optimization methods due to properties, such as high dimensionality, multimodality, epistasis (parameter interaction), non-differentiability, etc. a number of metaheuristics are developed over the years to search for viable solutions. While designing a metaheuristic, two contradictory criteria are taken into account, i.e., efficient exploration of the search space by preserving diversity (exploration) and exploitation of solutions found during a number of experiments while solving the same problem instance (exploitation). The metaheuristic must have the ability to visit the nonexplored regions of the search space without being confined to only a reduced number of regions. Further, during search of a metaheuristic, the promising regions are explored thoroughly to exploit already found better solutions. For satisfying both contradictory criteria, it is necessary to follow a random search guided by a decision-making strategy (probabilistic or population). Because of this randomness involved in the search strategy, these metaheuristics can provide sufficiently good solutions to an optimization problem or can analyze several problem instances with less computational effort as compared to numerical methods or simple heuristics or even a number of recently proposed metaheuristics in literature. The objective of this thesis work is to develop general purpose metaheuristics for performance analysis and design optimization of VLSI circuits. The proposed metaheuristics basically work on the philosophy of approximation and they tend to give satisfactory solutions in reasonable computational time.

In chapter 3, the problem of IR drop in power distribution networks is discussed and analyzed using the proposed methods. Usually the size of these power distribution networks are too big to be analyzed accurately and efficiently on a single computer due to lack of computing resources. Thus, for the efficient and accurate analysis of a power grid network, an appropriate methodology needs to be adopted along with the suitable computing environment. In view of this, two metaheuristics based on random walk (RW) and RFD are developed and described to analyze large power distribution networks. A parallel version of RW algorithm, i.e., TSRW is

proposed to efficiently detect hotspots on the power rails. The TSRW algorithm is designed to work on manycore platforms, e.g., GPU. TSRW has demonstrated $203\times$ speedup over serial RW algorithm and also outperforms state-of-the-art iterative and direct solvers while analyzing a power distribution network having 25 million nodes with less than 5% error. Although, this parallel metaheuristic is observed to show better performance, the error grows with the increase in size of power distribution networks. Therefore, a metaheuristic based on RFD is presented to analyze large power grid networks on GPU. The concept of RFD to pursue a path using gradient orientation is applied to identify hotspots across the power distribution network. Experimental results show that RFD outperforms RW and other iterative and direct solvers with acceptable accuracy loss (i.e, 3%) while solving a power distribution network having 25 million nodes.

The power distribution network analysis is performed in the later stage within the design flow to verify any signal integrity issues. Usually the designers over-constrain the design of power networks by extending their margins and routability to ensure reliability of power distribution network. This results in over-designing of the entire network making the overall area of design too congested for routing and placement of system design components. Therefore, to have a robust design, the power distribution network is redesigned considering signal integrity and wire area as cost functions. In chapter 4, the application of RFD method is extended to perform single objective optimization of power distribution networks. In this regard, the IR drop problem and the problem of excess wire area in power networks are transformed into two separate single objective optimization problems. Further, the two problems are analyzed using RFD and MRFD methods to come up with an optimum design of power distribution networks. It is observed that MRFD shows better performance in solving these two problems as compared to RFD and other peer algorithms.

With the aim of employing automation in the field of analog/RF circuit design, single objective optimization using RFD and MRFD methods developed in chapter 4 is extended to pursue analog/RF circuit optimization in chapter 5. At first, the analog circuit design

problem is transformed into a single objective optimization problem, where the circuit response is being formulated as an objective function subject to a variety of design constraints or other circuit specifications. Several experiments are performed on the two design examples to demonstrate the effectiveness in terms of applicability and solution quality of RFD and MRFD methods. The results are compared with other peer algorithms and it is observed that the two proposed metaheuristics show competitive performance. At second, the analog circuit design problem is formulated as a multiobjective optimization problem with the contradictory design specifications being represented as cost functions to be maximized or minimized. The tradeoff among these competing design specifications are analyzed by three proposed multiobjective optimization algorithms, i.e., hNSGA-II, MOMRFD and IMBSO. Several experiments are performed to showcase the tradeoff among different design specifications of a two-stage operational amplifier, a folded cascode amplifier and a low noise amplifier using proposed methodologies. It is observed that IMBSO has shown better performance than other peer algorithms and proposed methods in terms of diversity of Pareto optimal solutions. Since IMBSO makes use of brainstorming process as selection mechanism, the collective behavior of the improvements made in the IMBSO framework supports the diversity among cluster of individuals to evaluate Pareto optimal solutions.

As we know, CMOS-based technologies cannot keep up with the growing appeal for denser and low power circuits, e.g., the memory cell size is mainly limited by the size of access-transistor and the scaling of CMOS-based technology is also reaching its limit. Several factors force further scaling to die down to a limit, primarily increase in leakage current of access-transistors, slow growth in fabrication technology to keep pace with the scaling, etc. Emerging areas, e.g., MCAs can be viewed as promising alternatives which can offer ultra-small and low-power memory characteristics with fast switching speed. However, due to lower supply voltages and with the increase in size of MCAs, issues related to IR drop are likely to affect speed and endurance properties of such networks. In this case, MCAs need to be redesigned to place the supply PADs at appropriate locations across the MCAs. Therefore,

in chapter 6 of this thesis, we propose RFD and MRFD methods to minimize IR drop across large MCAs. The IR drop problem in MCA is transformed into a single objective minimization problem and the problem is solved to evaluate an optimum solution using the proposed methods successfully. Experimental results are compared with other peer algorithms and it is observed that MRFD has shown competitive performance than other peer algorithms in terms of convergence and quality of solution.

7.2 Directions for the future work

From the conclusion, it can be observed that the thesis is basically focused on design of efficient metaheuristic schemes for analysis and optimization of VLSI circuits. A few potential directions to which the contribution of the thesis can be extended, are discussed below.

- There is a scope to improve the performance of RFD in power distribution network analysis by efficient use of faster kernels on GPU or by partitioning the entire power distribution network and solving each part separately using efficient partitioning strategies [177]. The solutions found using RFD after power distribution network analysis can be provided as an input to iterative solvers, such as GMRES, PCG, etc. to improve the accuracy of these solvers while analyzing very large power distribution networks ($>$ 100 million nodes).

- The IR drop and area minimization problems can be translated into a multiobjective optimization problem, where IR drop and wire area can be regarded as two competing cost functions subjected to design constraints described in the thesis. In addition to these design constraints, additional constraints such as, timing yield, jitter, gate delay, etc., can also be considered in the design problem as constraints to improve performance and reliability of VLSI circuits.

- The power distribution network design optimization problem can be transformed into

a large scale variable optimization problem considering each metal width as a variable to be changed in each iteration while minimizing IR drop. In this case, cooperative co-evolution based strategies [178] can be employed to decompose the entire optimization problem into smaller subproblems and each subproblem can be solved using MRFD method to evaluate an optimum solution.

- Both metaheuristics and exact algorithms, such as integer programming techniques [102], etc. are complementary optimization strategies in terms of the quality of solutions and number of function evaluations. In the last few years, real-life optimization problems are evolved to be more complex and it leads to the development of efficient optimization software, libraries, and frameworks using hybrid approaches combining the metaheuristics and exact optimization algorithms or metaheuristics and nature-inspired methods for analyzing high-level modeling of such problem instances. In this regard, MRFD can be combined with other exact optimization algorithms or other metaheuristic schemes to develop a general purpose hybrid optimization framework.

- The multiobjective circuit optimization problem involving two or three objectives can be extended to solve more than three objective functions. While handling more than three objectives (many-objective optimization), the key challenge is to achieve balance between convergence and diversity. The proposed IMBSO framework can be extended to perform many-objective optimization by incorporating several dominance and decomposition based approaches [179, 180].

- The performance of the proposed MRFD metaheuristic on larger MCA benchmarks (more than 25 million nodes) can be studied in detail to showcase effectiveness compared to competing methods. In addition, the IR drop minimization of MCAs can be performed by incorporating a learning-based optimization framework to simultaneously train and redesign the crossbars.

A

Benchmarks for Power Distribution Network Analysis

Contents

A.1 Power distribution network benchmarks

The power distribution network (PDN) benchmarks for steady state analysis are produced in the SPICE format using our in-house power grid generator. The benchmark set includes five designs, i.e., pgnw1M, pgnw4M, pgnw9M, pgnw16M and pgnw25M. Table A.1 summarizes the important feature of these benchmarks. The table includes the benchmark name, number of nodes in the circuit representing the benchmark ($\#n$) and number of circuit elements (i.e., current sources ($\#i$), voltage sources ($\#v$) and resistors ($\#r$)). The metal resistances on these benchmarks are set to 1Ω similar to industrial designs. The V_{DD} PADs are placed randomly across the benchmarks to have a close realization of industrial power grid designs. The potential across these PADs are set to 1.8V and the current sinks are placed on each nodes reasonably with values set to 0.01A except at the PADs.

Table A.1: PDN benchmarks for steady state analysis

Benchmarks	#n	#i	#r	#v
pgnw1M	1000001	998001	1998000	2000
pgnw4M	4000001	3996001	7996000	4000
pgnw9M	9000001	8994001	17994000	6000
pgnw16M	16000001	15992001	31992000	8000
pgnw25M	25000001	24990001	49990000	10000

Table A.2: PDN benchmarks for transient analysis

Benchmarks	#n	#i	#r	#v	#c
pgtnw10K	10001	9800	19800	200	9800
pgtnw20K	19882	19600	39480	282	19600
pgtnw30K	29930	29584	59512	346	29584
pgtnw40K	40001	39600	79600	400	39600
pgtnw50K	49730	49284	99012	446	49284
pgtnw100K	99857	99224	199080	632	99224
pgtnw500K	499850	498436	998284	1414	498436
pgtnw1M	1000001	998000	1998000	2000	998000
pgtnw4M	4000001	3996000	7996000	4000	3996000

Similarly, the benchmarks for transient analysis are also produced in the SPICE format with a grid size ranging from ten thousand to 4 million nodes, i.e., pgtnw10K to pgtnw4M.

Table A.2 lists the important features of these benchmarks, i.e., the benchmark name, number of nodes in the circuit representing the benchmark ($\#n$) and number of circuit elements (i.e, current sources ($\#i$), voltage sources ($\#v$), resistors ($\#r$) and capacitors ($\#c$)). The metal resistances on these benchmarks are set to 1Ω and capacitances are set to 0.1μF similar to industrial designs. V_{DD} PADs having potential set to 1.8V are placed randomly across the benchmarks to have a close realization of industrial power grid designs. The current sinks are placed on each nodes reasonably with values set to 0.01A except at the PADs.

B

Lower Bound of Constriction Coefficient

Contents

B.1 Lower bound of constriction coefficient

For $D > 0$, considering $V > R$, the constriction coefficient w described in chapter 4, can be evaluated by solving (B.1).

$$\begin{aligned}
&-\frac{1}{9}(w^2+4w+1) > -\frac{1}{27}(13w^2+4w+1)(1+2w)\\
\Rightarrow &(w^2+4w+1) < \frac{1}{3}(13w^2+4w+1)(1+2w)\\
\Rightarrow &(1+2w)^2 - 3w^2 < \frac{1}{3}(13w^2+4w+1)(1+2w)\\
\Rightarrow &(1+2w) - \frac{3w^2}{1+2w} < \frac{1}{3}(13w^2+4w+1)\\
\Rightarrow &-\frac{3w^2}{1+2w} - \frac{13w^2}{3} + \frac{2w}{3} + \frac{2}{3} < 0\\
\Rightarrow &13w^3 + 9w^2 - 3w - 1 > 0
\end{aligned} \tag{B.1}$$

Further, solving (B.1), it can be shown that all roots are real and one of the roots has a lower bound, $w > 1$.

C

Multiobjective test functions and performance of proposed optimization techniques

Contents

C.1 Standard Multiobjective Test Functions

We choose to demonstrate the effectiveness of proposed multiobjective optimization algorithms by finding solutions to different sets of multi-objective test functions. These test functions have been used as standard benchmarks in literature. A total of twelve test problems with two different groups of benchmarks are considered (both constrained and unconstrained test problems). Details on these test functions are listed in Table C.1.

C.2 Quality Indicators

The quantitative comparison of the performance of different multiobjective optimization algorithms is usually performed with respect to the Pareto optimal set. For evaluating the quality of solutions, two factors are considered in each evaluation metric, i.e., closeness of solutions and the extent of diversity among solutions with respect to true Pareto front. Various performance metrics are reported in this regard to analyze the solutions obtained using different multiobjective optimization algorithms. These performance metrics are often referred as quality indicators. The quality indicators, considered in the thesis are given below.

C.2.1 Generational Distance (GD)

Generational distance evaluates the average distance of solutions from each point in the obtained Pareto front to the nearest point of solutions on a true Pareto front and it can be expressed as follows [97],

$$\text{GD} = \frac{\sqrt{\sum_{i=1}^{N_{obt}} (d_i)^2}}{N_{obt}}, \tag{C.1}$$

where, N_{obt} represents the number of solutions in obtained Pareto front and d_i represents the Euclidean distance between the i^{th} obtained solution and the closest point of the true Pareto front.

Table C.1: Standard multiobjective test functions

Benchmark	Test Function	Constraints	Search Domain
ZDT1	$Minimize = \begin{cases} f_1(x) = x_1 \\ f_2(x) = g(x)h(f_1(x), g(x)) \\ g(x) = 1 + \frac{9}{29}\sum_{i=2}^{30} x_i \\ h(f_1(x), g(x)) = 1 - \sqrt{\frac{f_1(x)}{g(x)}} \end{cases}$		$0 \leq x_i \leq 1$ $1 \leq i \leq 30$
ZDT2	$Minimize = \begin{cases} f_1(x) = x_1 \\ f_2(x) = g(x)h(f_1(x), g(x)) \\ g(x) = 1 + \frac{9}{29}\sum_{i=2}^{30} x_i \\ h(f_1(x), g(x)) = 1 - (\frac{f_1(x)}{g(x)})^2 \end{cases}$		$0 \leq x_i \leq 1$ $1 \leq i \leq 30$
ZDT3	$Minimize = \begin{cases} f_1(x) = x_1 \\ f_2(x) = g(x)h(f_1(x), g(x)) \\ g(x) = 1 + \frac{9}{29}\sum_{i=2}^{30} x_i \\ h(f_1(x), g(x)) = 1 - \sqrt{\frac{f_1(x)}{g(x)}} - (\frac{f_1(x)}{g(x)})sin(10\pi f_1(x)) \end{cases}$		$0 \leq x_i \leq 1$ $1 \leq i \leq 30$
ZDT4	$Minimize = \begin{cases} f_1(x) = x_1 \\ f_2(x) = g(x)h(f_1(x), g(x)) \\ g(x) = 1 + \sum_{i=2}^{10}(x_i^2 - 10cos(4\pi x_i)) \\ h(f_1(x), g(x)) = 1 - \sqrt{\frac{f_1(x)}{g(x)}} \end{cases}$		$0 \leq x_1 \leq 1$ $-5 \leq x_i \leq 5$ $2 \leq i \leq 10$
ZDT6	$Minimize = \begin{cases} f_1(x) = x_1 \\ f_2(x) = g(x)h(f_1(x), g(x)) \\ g(x) = 1 + \frac{9}{29}\sum_{i=2}^{30} x_i \\ h(f_1(x), g(x)) = 1 - (\frac{f_1(x)}{g(x)})^2 \end{cases}$		$0 \leq x_i \leq 1$ $1 \leq i \leq 10$
SCH	$Minimize = \begin{cases} f_1(x) = x^2 \\ f_2(x) = (x-2)^2 \end{cases}$		$0 \leq x_i \leq 1$ $1 \leq i \leq 30$
FON	$Minimize = \begin{cases} f_1(x) = 1 - exp[-\sum_{i=1}^{n}(x_i - \frac{1}{\sqrt{n}})^2] \\ f_2(x) = 1 - exp[-\sum_{i=1}^{n}(x_i + \frac{1}{\sqrt{n}})^2] \end{cases}$		$-4 \leq x_i \leq 4$ $1 \leq i \leq n$
KUR	$Minimize = \begin{cases} f_1(x) = \sum_{i=1}^{2}[-10exp(-0.2\sqrt{x_i^2 + x_{i+1}^2})] \\ f_2(x) = \sum_{i=1}^{3}[\lvert x_i\rvert^{.8} + 5sin(x_i^3)] \end{cases}$		$-5 \leq x_i \leq 5$ $1 \leq i \leq 3$
BIN	$Minimize = \begin{cases} f_1(x,y) = 4x^2 + 4y^2 \\ f_2(x,y) = (x-5)^2 + (y-5)^2 \end{cases}$	$g_1(x,y) = (x-5)^2 + y^2 \leq 25,$ $g_2(x,y) = (x-8)^2 + (y+3)^2 \geq 7.7$	$0 \leq x \leq 5$ $0 \leq y \leq 3$
CEX	$Minimize = \begin{cases} f_1(x,y) = x \\ f_2(x,y) = \frac{1+y}{x} \end{cases}$	$g_1(x,y) = y + 9x \geq 6,$ $g_2(x,y) = -y + 9x \geq 1$	$0.1 \leq x \leq 1$ $0 \leq y \leq 5$
OZY	$Minimize = \begin{cases} f_1(x) = -25(x_1-2)^2 - (x_2-2)^2 - (x_3-1)^2 - (x_4-4)^2 - (x_5-1)^2 \\ f_2(x) = \sum_{i=1}^{6} x_i^2 \end{cases}$	$g_1(x) = x_1 + x_2 - 2 \geq 0,\ g_2(x) = 6 - x_1 - x_2 \geq 0$ $g_3(x) = 2 - x_2 + x_1 \geq 0,\ g_4(x) = 2 - x_1 + 3x_2 \geq 0,$ $g_5(x) = 4 - (x_3-3)^2 - x_4 \geq 0,$ $g_6(x) = (x_5-3)^2 + x_6 - 4 \geq 0$	$0 \leq x_1, x_{2,x} 6 \leq 10$ $1 \leq x_3, x_5 \leq 5$ $0 \leq x_4 \leq 6$
TNK	$Minimize = \begin{cases} f_1(x) = x_1 \\ f_2(x) = x_2 \end{cases}$	$g_1(x) = -x_1^2 - x_2^2 + 1 + 0.1cos(16arctan(\frac{x_1}{x_2})) \leq 0,$ $g_2(x) = (x_1 - .5)^2 + (x_2 - .5)^2 \leq 0.5$	$0 \leq x_i \leq \pi$ $1 \leq i \leq 2$

C.2.2 Inverted Generational Distance (IGD)

The inverted generational distance is an assessment criterion, which evaluates the average distance form each point of true Pareto front to the nearest point in obtained Pareto front. It can be expressed as follows [97],

$$IGD = \frac{\sqrt{\sum_{i=1}^{N_{true}} (d_i')^2}}{N_{true}}, \tag{C.2}$$

where, N_{true} is the number of solutions on true Pareto front, and d_i' represents the Euclidean distance between the i^{th} solutions on the true Pareto front and the nearest solution on the obtained Pareto front.

C.2.3 Hypervolume (HV)

Hyervolume metric is used to measure convergence and diversity of optimization algorithms. HV measures the volume or area of the dominated portion of the objective space, which is covered by the obtained Pareto optimal solutions. Assuming a point x_i in $\mathcal{P}_{obt}$ with a rectangular area, $a(x_i)$, which is bounded by origin and the point x_i . The union of such rectangular areas is referred as hyperarea (hypervolume in the case of 3D Pareto fronts), and in general it is called hypervolume of $\mathcal{P}_{obt}$, which can be expressed as [97],

$$HF(\mathcal{P}_{obt}) = \left\{ \bigcup_{i}^{n} a(x_i) | \forall x_i \in \mathcal{P}_{obt} \right\} \tag{C.3}$$

The hyperarea ratio metric can also be expressed as [97],

$$HR = \frac{HF(\mathcal{P}_{obt})}{HF(\mathcal{P}_{true})} \tag{C.4}$$

where, $HF(\mathcal{P}_{true})$ is hyperarea or hypervolume covered by true Pareto front of the function. HV metric provides a qualitative measure of convergence as well as diversity in a combined manner. Nevertheless, it can be used along with GD or IGD metrics to get a better overview of the performance of algorithm.

C.3 Performance Assessment

With the aim of giving a complete overview of performance of hNSGA-II, MOMRFD and IMBSO, several standard optimization algorithms (i.e, NSGA-II, SPEA-II, dMOPSO, SMPSO) are considered for evaluation on all twelve benchmarks (ZDT1, ZDT2, ZDT3, ZDT4, ZDT6, SCH, FON, KUR, BIN, CEX, OZY and TNK [7]). For comparison, we have used the authors implementation of all the algorithms. During evaluation of all the algorithms, a population size of 200 is considered to evolve over 1000 generations. Different parameters are set as listed in Table 5.11 of chapter 5.

It is observed from Table C.1 that GD values are better for five out of twelve benchmarks for the proposed IMBSO method. This indicates that the resulting Pareto fronts from IMBSO are closer to the true Pareto fronts in these five benchmark functions. From Table C.1, it can be seen that IMBSO obtains lowest values fro the IGD metric in ten out of twelve metrics. This points out that the obtained Pareto fronts of multi-objective test functions are closer to true Pareto fronts and the nondominated solutions are better spread for IMBSO as compared to other algorithms. Further, to showcase the performance of IMBSO as a measure of both convergence and diversity in a concise manner, HV metric is evaluated for all benchmarks. It can be seen from Table C.1 that IMBSO obtains best (highest) HV values in all twelve benchmarks. In response to the results, it can be concluded that IMBSO shows better performance than other peer algorithms and the proposed methods (hNSGA-II and MOMRFD) in terms of convergence and diversity of solutions.

Table C.2: Mean values of GD, IGD and HV metrics of different algorithms on standard test functions

Methods	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6	SCH	FON	KUR	BIN	CEX	TNK	OZY
GD metric:												
NSGA-II	1.1517e-01	2.0507e-01	9.6721e-02	4.0117e+01	0.2416e+01	1.1460e-02	2.4156e-03	7.3262e-02	1.8161e+01	6.0349e-03	1.1905e-02	0.2725e+01
SPEA-II	3.4137e-03	7.7292e-03	1.3506e-03	1.9055e-01	1.1563e-01	4.3401e-01	2.6145e-04	2.4346e-04	3.2894e-02	2.5679e-04	1.7636e-03	5.9922e-04
dMOPSO	1.0490e-04	4.9559e-05	9.8536e-05	8.9505e-05	4.1801e-05	2.2700e-04	1.2825e-04	4.1136e-04	9.6670e-03	1.8696e-02	5.470e+01	0.9520e+01
SMPSO	1.0256e-04	4.8441e-05	1.1824e-04	7.4796e-05	1.6137e-03	2.2366e-04	1.2841e-04	2.4206e-04	2.7278e-02	2.0802e-04	1.6630e-03	9.0512e-04
*h*NSGA-II	1.4490e-03	2.3122e-03	1.6974e-03	0.2816e+01	2.0486e-01	1.0514e-02	1.2167e-03	4.3495e-03	1.9121e-01	7.2328e-03	8.1449e-05	2.2431e-02
MOMRFD	1.6441e-03	9.8105e-04	1.6855e-03	2.1120e+01	1.1098e-02	1.0874e-02	1.1867e-03	3.4030e-02	2.0189e+01	6.6144e-03	1.4894e-02	3.7541e-03
IMBSO	1.0401e-04	2.1578e-05	2.5601e-05	7.8650e-05	2.1421e-04	1.4523e-04	2.0145e-04	3.6124e-05	1.5012e-03	2.1031e-04	6.2142e-04	3.1254e-03
IGD metric:												
NSGA-II	7.9356e-02	1.4510e-01	7.2018e-03	1.4407e+01	8.4401e-01	2.2365e-03	7.8359e-04	1.2120e-02	3.8746e-02	2.6953e-03	1.3635e-03	6.1283e-01
SPEA-II	1.0155e-03	1.5281e-02	5.7333e-04	3.3890e-02	1.3288e-02	3.1697e-02	2.3973e-04	1.3727e-04	1.5887e-04	5.0045e-04	8.8447e-03	6.1796e-04
dMOPSO	1.5926e-04	1.4076e-04	2.8151e-04	1.6759e-04	1.3987e-04	1.3766e-03	2.0804e-04	2.2666e-04	1.2544e-04	8.2293e-03	1.2335e-02	8.0866e-02
SMPSO	1.3510e-04	1.3807e-04	1.0258e-04	1.3529e-04	1.3442e-04	3.5439e-04	2.4157e-04	1.5296e-04	1.1615e-04	1.9699e-04	1.0219e-02	6.2999e-04
*h*NSGA-II	5.3687e-04	2.3605e-03	8.0021e-04	0.2497e+01	2.2087e-02	2.0502e-03	5.3678e-04	5.1619e-03	4.3515e-02	2.4158e-03	9.5842e-04	2.1583e-01
MOMRFD	8.6492e-04	9.6824e-04	1.6504e-03	1.1024e-01	2.4058e-04	1.6504e-03	4.2177e-04	4.6057e-03	3.7811e-02	1.7254e-03	9.0366e-04	3.4017e-03
IMBSO	1.0142e-05	1.2507e-05	1.0784e-04	1.1453e-04	1.1578e-05	3.1241e-05	5.8910e-05	1.4511e-04	1.6542e-05	1.8902e-05	2.1403e-04	2.1455e-04
HV metric:												
NSGA-II	0.6575	0.3241	0.5131	0.6518	0.3773	0.6620	0.3079	0.3994	0.7249	0.7629	0.3343	0.3621
SPEA-II	0.6164	0.0528	0.4868	0.6577	0.0447	0.5504	0.3102	0.3996	0.7242	0.7752	0.7045	0.3047
dMOPSO	0.6615	0.3283	0.5136	0.6609	0.4013	0.8284	0.3118	0.3962	0.7283	0.8305	0.7567	0.7194
SMPSO	0.6617	0.3286	0.5155	0.6615	0.4012	0.8208	0.3124	0.4001	0.7275	0.7756	0.6981	0.3047
*h*NSGA-II	0.6667	0.4319	0.7818	0.6489	0.2188	0.6618	0.3051	0.4933	0.7509	0.7380	0.6512	0.4526
MOMRFD	0.6801	0.6278	0.7152	0.7021	0.4035	0.7862	0.6131	0.6971	0.7771	0.7809	0.7825	0.4831
IMBSO	0.9978	0.9931	0.9981	0.9975	0.9963	0.9982	0.9826	0.9752	0.9727	0.9779	0.8602	0.7061

Bibliography

[1] G. E. Moore, "Cramming more components onto integrated circuits, reprinted from electronics, volume 38, number 8, april 19, 1965, pp. 114 ff." *IEEE solid-state circuits society newsletter*, vol. 20, no. 3, pp. 33–35, 2006.

[2] A. Mezhiba and E. G. Friedman, *Power distribution networks in high speed integrated circuits*. Springer Science & Business Media, 2012.

[3] A. Dharchoudhury, R. Panda, D. Blaauw, R. Vaidyanathan, B. Tutuianu, and D. Bearden, "Design and analysis of power distribution networks in powerpc microprocessors," in *Proceedings of the 35th annual Design Automation Conference*. ACM, 1998, pp. 738–743.

[4] A. Allan, D. Edenfeld, W. H. Joyner, A. B. Kahng, M. Rodgers, and Y. Zorian, "2001 technology roadmap for semiconductors," *Computer*, vol. 35, no. 1, pp. 42–53, 2002.

[5] Y.-M. Lee and C.-T. Ho, "Intrasim: Incremental transient simulation of power grids," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2017.

[6] V. Mishra and S. S. Sapatnekar, "The impact of electromigration in copper interconnects on power grid integrity," in *Design Automation Conference (DAC), 2013 50th ACM/EDAC/IEEE*. IEEE, 2013, pp. 1–6.

[7] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[8] A. Steane, "Quantum computing," *Reports on Progress in Physics*, vol. 61, no. 2, p. 117, 1998.

[9] D. D. Awschalom and M. E. Flatté, "Challenges for semiconductor spintronics," *Nature Physics*, vol. 3, no. 3, pp. 153–159, 2007.

[10] Z. Yin, H. Tian, G. Chen, and L. O. Chua, "What are memristor, memcapacitor, and meminductor?" *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 62, no. 4, pp. 402–406, 2015.

[11] B. Liu, W. Wen, Y. Chen, X. Li, C.-R. Wu, and T.-Y. Ho, "Eda challenges for memristor-crossbar based neuromorphic computing," in *Proceedings of the 25th edition on Great Lakes Symposium on VLSI*. ACM, 2015, pp. 185–188.

[12] G. J. Myers, *Advances in computer architecture*. John Wiley & Sons, Inc., 1982.

[13] B. Liu, H. Li, Y. Chen, X. Li, T. Huang, Q. Wu, and M. Barnell, "Reduction and ir-drop compensations techniques for reliable neuromorphic computing systems," in *2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2014, pp. 63–70.

[14] M. del Mar Hershenson, S. P. Boyd, and T. H. Lee, "Gpcad: A tool for cmos op-amp synthesis," in *Proceedings of the 1998 IEEE/ACM international conference on Computer-aided design*. ACM, 1998, pp. 296–303.

[15] *Analog Design Automation, Inc.*, Available: ¡http://www.analogda.com¿ Std.

[16] *Cadence Inc., Products: Composer, Virtuoso, DIVA, NeoCircuit, NeoCell, UltraSim, NcSim.*, Available: ¡http://www.cadence.com¿ Std.

[17] L. Nagel, "Spice 2 : A computer program to simulate semiconductor circuits," Ph.D. dissertation, University of California, Berkeley, May 1978.

[18] G. G. Gielen and R. A. Rutenbar, "Computer-aided design of analog and mixed-signal integrated circuits," *Proceedings of the IEEE*, vol. 88, no. 12, pp. 1825–1854, 2000.

[19] Q. Zhou, K. Sun, K. Mohanram, and D. Sorensen, "Large power grid analysis using domain decomposition," in *Design, Automation and Test in Europe, 2006. DATE '06. Proceedings*, vol. 1, March 2006, pp. 1–6.

[20] T. M. et al, "Fast and memory-efficient gpu implementations of krylov subspace methods for efficient power grid analysis," in *GLSVLSI'13*, May 23, 2013, Paris, France.

[21] S. S. S. Haifeng Qian, "Hierarchical random-walk algorithms for power grid analysis," *Proceedings of the Asia and South Pacific Design Automation Conference*, 2004.

[22] S. N. J. Kozhaya and F. Najm, "A multigrid-like technique for power grid analysis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 21, pp. 1148–1160, 2002.

[23] M. Z. Cheng Zhuo, Jiang Hu and K. Chen, "Power grid analysis and optimization using algebraic multigrid," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. VOL. 27, pp. 738–751, Apr. 2008.

[24] J. Yang, Y. Cai, Q. Zhou, and J. Shi, "Fast poisson solver preconditioned method for robust power grid analysis," in *ICCAD*, NJ, USA, 2011, pp. 531–536.

[25] F. N. Najm *et al.*, "Incremental power grid verification," in *Proceedings of the 49th Annual Design Automation Conference*. ACM, 2012, pp. 151–156.

[26] T. Y. Wang and C. P. Chen, "Optimization of the power/ground network wire-sizing and spacing based on sequential network simplex algorithm," in *Quality Electronic Design, 2002. Proceedings. International Symposium on*. IEEE, 2002, pp. 157–162.

[27] K. Wang and M. Marek-Sadowska, "On-chip power-supply network optimization using multigrid-based technique," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 3, pp. 407–417, 2005.

[28] H. Su, S. S. Sapatnekar, and S. R. Nassif, "Optimal decoupling capacitor sizing and placement for standard-cell layout designs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 4, pp. 428–436, 2003.

[29] D. Stringfellow and J. Pedicone, "Decoupling capacitance estimation, implementation, and verification: A practical approach for deep submicron socs," *Synopsys Users Group*, 2007.

[30] X.-D. S. Tan and C.-J. R. Shi, "Fast power/ground network optimization based on equivalent circuit modeling," in *Proceedings of the 38th annual Design Automation Conference*. ACM, 2001, pp. 550–554.

[31] X. Wu, X. Hon, Y. Ca, C.-K. Cheng, J. Gu, and W. Dai, "Area minimization of power distribution network using efficient nonlinear programming techniques," in *Computer Aided Design, 2001. ICCAD 2001. IEEE/ACM International Conference on*. IEEE, 2001, pp. 153–157.

[32] S. X.-D. Tan, C.-J. R. Shi, and J.-C. Lee, "Reliability-constrained area optimization of vlsi power/ground networks via sequence of linear programmings," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 12, pp. 1678–1684, 2003.

[33] T. Mitsuhashi and E. S. Kuh, "Power and ground network topology optimization for cell based vlsis," in *Proceedings of the 29th ACM/IEEE Design Automation Conference*. IEEE Computer Society Press, 1992, pp. 524–529.

[34] R. Fletcher, *Practical methods of optimization*. John Wiley & Sons, 2013.

[35] M. Zhao, Y. Fu, V. Zolotov, S. Sundareswaran, and R. Panda, "Optimal placement of power-supply pads and pins," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 1, pp. 144–154, 2006.

[36] Y. Zhong and M. D. Wong, "Fast placement optimization of power supply pads," in *Design Automation Conference, 2007. ASP-DAC'07. Asia and South Pacific*. IEEE, 2007, pp. 763–767.

[37] G. Gielen, H. Walscharts, and W. Sansen, "Analog circuit design optimization based on symbolic simulation and simulated annealing," in *Solid-State Circuits Conference, 1989. ESSCIRC '89. Proceedings of the 15th European*, Sept 1989, pp. 252–255.

[38] H. Liu, A. Singhee, R. A. Rutenbar, and L. R. Carley, "Remembrance of circuits past: macromodeling by data mining in large analog design spaces," in *Proceedings of the 39th annual Design Automation Conference*. ACM, 2002, pp. 437–442.

[39] 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, 2003.

[40] S. P. Boyd, T. H. Lee *et al.*, "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, 2001.

[41] 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, 1989.

[42] 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.

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

[44] I. Guerra-Gómez, E. Tlelo-Cuautle, T. McConaghy, and G. Gielen, "Optimizing current conveyors by evolutionary algorithms including differential evolution," in *Electronics, Circuits, and Systems, 2009. ICECS 2009. 16th IEEE International Conference on*. IEEE, 2009, pp. 259–262.

[45] T. O. Weber and W. A. Van Noije, "Analog design synthesis method using simulated annealing and particle swarm optimization," in *Proceedings of the 24th symposium on Integrated circuits and systems design*. ACM, 2011, pp. 85–90.

[46] L. Chua, "Memristor-the missing circuit element," *IEEE Transactions on circuit theory*, vol. 18, no. 5, pp. 507–519, 1971.

[47] A. Bachtold, P. Hadley, T. Nakanishi, and C. Dekker, "Logic circuits with carbon nanotube transistors," *Science*, vol. 294, no. 5545, pp. 1317–1320, 2001.

[48] A. Javey, J. Guo, Q. Wang, M. Lundstrom, and H. Dai, "Ballistic carbon nanotube field-effect transistors," *nature*, vol. 424, no. 6949, p. 654, 2003.

[49] S. Wolf, D. Awschalom, R. Buhrman, J. Daughton, S. Von Molnar, M. Roukes, A. Y. Chtchelkanova, and D. Treger, "Spintronics: a spin-based electronics vision for the future," *Science*, vol. 294, no. 5546, pp. 1488–1495, 2001.

[50] J. J. Yang, D. B. Strukov, and D. R. Stewart, "Memristive devices for computing," *Nature nanotechnology*, vol. 8, no. 1, pp. 13–24, 2013.

[51] J. H. Holland, "Genetic algorithms and the optimal allocation of trials," *SIAM Journal on Computing*, vol. 2, no. 2, pp. 88–105, 1973.

[52] K. Deb and D. Deb, "Analysing mutation schemes for real-parameter genetic algorithms," *International Journal of Artificial Intelligence and Soft Computing*, vol. 4, no. 1, pp. 1–28, 2014.

[53] R. Poli and W. B. Langdon, "On the search properties of different crossover operators in genetic programming," *Genetic Programming*, pp. 293–301, 1998.

[54] R. Cheng, M. Gen, and Y. Tsujimura, "A tutorial survey of job-shop scheduling problems using genetic algorithmsi. representation," *Computers & industrial engineering*, vol. 30, no. 4, pp. 983–997, 1996.

[55] J. Grefenstette, R. Gopal, B. Rosmaita, and D. Van Gucht, "Genetic algorithms for the traveling salesman problem," in *Proceedings of the first International Conference on Genetic Algorithms and their Applications*. Lawrence Erlbaum, New Jersey (160-168), 1985, pp. 160–168.

[56] C. Changdar, G. Mahapatra, and R. K. Pal, "An efficient genetic algorithm for multi-objective solid travelling salesman problem under fuzziness," *Swarm and Evolutionary Computation*, vol. 15, pp. 27 – 37, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2210650213000679

[57] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," *Machine learning*, vol. 3, no. 2, pp. 95–99, 1988.

[58] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.

[59] K. Price, R. M. Storn, and J. A. Lampinen, *Differential evolution: a practical approach to global optimization*. Springer Science & Business Media, 2006.

[60] R. Mallipeddi, P. N. Suganthan, Q.-K. Pan, and M. F. Tasgetiren, "Differential evolution algorithm with ensemble of parameters and mutation strategies," *Applied soft computing*, vol. 11, no. 2, pp. 1679–1696, 2011.

[61] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE transactions on evolutionary computation*, vol. 15, no. 1, pp. 4–31, 2011.

[62] 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, 1997.

[63] A. K. Qin and P. N. Suganthan, "Self-adaptive differential evolution algorithm for numerical optimization," in *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, vol. 2. IEEE, 2005, pp. 1785–1791.

[64] J. Zhang and A. C. Sanderson, "Jade: adaptive differential evolution with optional external archive," *IEEE Transactions on evolutionary computation*, vol. 13, no. 5, pp. 945–958, 2009.

[65] R. Tanabe and A. S. Fukunaga, "Improving the search performance of shade using linear population size reduction," in *Evolutionary Computation (CEC), 2014 IEEE Congress on*. IEEE, 2014, pp. 1658–1665.

[66] R. Poláková, "L-shade with competing strategies applied to constrained optimization," in *Evolutionary Computation (CEC), 2017 IEEE Congress on*. IEEE, 2017, pp. 1683–1689.

[67] D. Bratton and J. Kennedy, "Defining a standard for particle swarm optimization," in *Swarm Intelligence Symposium, 2007. SIS 2007. IEEE*. IEEE, 2007, pp. 120–127.

[68] Q. Liu, "Order-2 stability analysis of particle swarm optimization," *Evolutionary computation*, vol. 23, no. 2, pp. 187–216, 2015.

[69] Y. Zhang, S. Wang, and G. Ji, "A comprehensive survey on particle swarm optimization algorithm and its applications," *Mathematical Problems in Engineering*, vol. 2015, 2015.

[70] L. dos Santos Coelho, "Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems," *Expert Systems with Applications*, vol. 37, no. 2, pp. 1676–1683, 2010.

[71] B. Liu, L. Wang, Y.-H. Jin, F. Tang, and D.-X. Huang, "Improved particle swarm optimization combined with chaos," *Chaos, Solitons & Fractals*, vol. 25, no. 5, pp. 1261–1271, 2005.

[72] A. M. Abdelbar, S. Abdelshahid, and D. C. Wunsch, "Fuzzy pso: a generalization of particle swarm optimization," in *Neural Networks, 2005. IJCNN'05. Proceedings. 2005 IEEE International Joint Conference on*, vol. 2. IEEE, 2005, pp. 1086–1091.

[73] S. Ho, S. Yang, G. Ni, E. W. Lo, and H.-c. C. Wong, "A particle swarm optimization-based method for multiobjective design optimizations," *IEEE transactions on magnetics*, vol. 41, no. 5, pp. 1756–1759, 2005.

[74] P. Upadhyay, R. Kar, D. Mandal, and S. Ghoshal, "Craziness based particle swarm optimization algorithm for iir system identification problem," *AEU-International Journal of Electronics and Communications*, vol. 68, no. 5, pp. 369–378, 2014.

[75] D. Joshi, S. Dash, U. Agarwal, R. Bhattacharjee, and G. Trivedi, "Analog circuit optimization based on hybrid particle swarm optimization," in *2015 International Conference on Computational Science and Computational Intelligence (CSCI)*, Dec 2015, pp. 164–169.

[76] Y. Hariya, T. Kurihara, T. Shindo, and K. Jin'no, "Lévy flight pso," in *Evolutionary Computation (CEC), 2015 IEEE Congress on*. IEEE, 2015, pp. 2678–2684.

[77] H. Haklı and H. Uğuz, "A novel particle swarm optimization algorithm with levy flight," *Applied Soft Computing*, vol. 23, pp. 333–345, 2014.

[78] W.-N. Chen, J. Zhang, Y. Lin, N. Chen, Z.-H. Zhan, H. S.-H. Chung, Y. Li, and Y.-H. Shi, "Particle swarm optimization with an aging leader and challengers," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 2, pp. 241–258, 2013.

[79] C. A. C. Coello, G. T. Pulido, and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE Transactions on evolutionary computation*, vol. 8, no. 3, pp. 256–279, 2004.

[80] A. Nebro, J. Durillo, J. García-Nieto, C. Coello Coello, F. Luna, and E. Alba, "Smpso: A new pso-based metaheuristic for multi-objective optimization," in *2009 IEEE Symposium on Computational Intelligence in Multicriteria Decision-Making (MCDM 2009)*. IEEE Press, 2009, pp. 66–73.

[81] S. Z. Martínez and C. A. C. Coello, "A multi-objective particle swarm optimizer based on decomposition," in *GECCO*, 2011, pp. 69–76.

[82] S. N. H. Qian and S. Sapatnekar, "Power grid analysis using random walks," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, pp. 1204–1224, 2005.

[83] Z. L. Weikun GUO, Sheldon X. D. TAN and X. HANG, "Partial random walks for transient analysis of large power distribution networks," *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences*, vol. Vol.E87-A , No.12, pp. pp.3265–3272, 2004.

[84] P. Rabanal, I. Rodriguez, and F. Rubio, "Solving dynamic tsp by using river formation dynamics," in *Natural Computation, 2008. ICNC '08. Fourth International Conference on*, vol. 1, Oct 2008, pp. 246–250.

[85] P. Rabanal, I. Rodríguez, and F. Rubio, "Applications of river formation dynamics," *Journal of Computational Science*, vol. 22, pp. 26–35, 2017.

[86] S. Mehrjoo and F. Khunjush, "Optimal data aggregation tree in wireless sensor networks based on improved river formation dynamics," *Computational Intelligence*, 2017.

[87] K. Miettinen, *Nonlinear multiobjective optimization*. Springer Science & Business Media, 2012, vol. 12.

[88] Y. Shi, J. Xue, and Y. Wu, "Multi-objective optimization based on brain storm optimization algorithm," *International Journal of Swarm Intelligence Research (IJSIR)*, vol. 4, no. 3, pp. 1–21, 2013.

[89] Z.-h. Zhan, J. Zhang, Y.-h. Shi, and H.-l. Liu, "A modified brain storm optimization," in *Evolutionary Computation (CEC), 2012 IEEE Congress on*. IEEE, 2012, pp. 1–8.

[90] D. Zhou, Y. Shi, and S. Cheng, "Brain storm optimization algorithm with modified step-size and individual generation," *Advances in swarm intelligence*, pp. 243–252, 2012.

[91] Y. Shi, "Brain storm optimization algorithm in objective space," in *Evolutionary Computation (CEC), 2015 IEEE Congress on*. IEEE, 2015, pp. 1227–1234.

[92] C. Sun, H. Duan, and Y. Shi, "Optimal satellite formation reconfiguration based on closed-loop brain storm optimization," *IEEE Computational Intelligence Magazine*, vol. 8, no. 4, pp. 39–51, 2013.

[93] H. Duan, S. Li, and Y. Shi, "Predator–prey brain storm optimization for dc brushless motor," *IEEE Transactions on Magnetics*, vol. 49, no. 10, pp. 5336–5340, 2013.

[94] H. Qiu, H. Duan, and Y. Shi, "A decoupling receding horizon search approach to agent routing and optical sensor tasking based on brain storm optimization," *Optik-International Journal for Light and Electron Optics*, vol. 126, no. 7, pp. 690–696, 2015.

[95] S. García, D. Molina, M. Lozano, and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms behaviour: a case study on the cec2005 special session on real parameter optimization," *Journal of Heuristics*, vol. 15, no. 6, p. 617, 2009.

[96] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.

[97] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Transactions on evolutionary computation*, vol. 7, no. 2, pp. 117–132, 2003.

[98] S. Jiang, Y.-S. Ong, J. Zhang, and L. Feng, "Consistencies and contradictions of performance metrics in multiobjective optimization," *IEEE transactions on cybernetics*, vol. 44, no. 12, pp. 2391–2404, 2014.

[99] X. Cai, H. Sun, and Z. Fan, "A diversity indicator based on reference vectors for many-objective optimization," *Information Sciences*, vol. 430, pp. 467–486, 2018.

[100] C. Grosan and A. Abraham, "Hybrid evolutionary algorithms: methodologies, architectures, and reviews," in *Hybrid evolutionary algorithms*. Springer, 2007, pp. 1–17.

[101] C. Blum, J. Puchinger, G. R. Raidl, and A. Roli, "Hybrid metaheuristics in combinatorial optimization: A survey," *Applied Soft Computing*, vol. 11, no. 6, pp. 4135–4151, 2011.

[102] G. R. Raidl and J. Puchinger, "Combining (integer) linear programming techniques and metaheuristics for combinatorial optimization," in *Hybrid metaheuristics*. Springer, 2008, pp. 31–62.

[103] H. L. Yu H. and T. S., "Bsmor: Block structure preserving model order reduction," in *IEEE International BMAS*, Sep. 2005.

[104] S. J.M. and L. Silveira, "On the compressibility of power grid models," in *IEEE ISVLSI '07*, Mar 2007.

[105] S. S. M. Zhao, R. V. Panda and D. Blaauw, "Hierarchical analysis of power distribution networks," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. vol. 21, no. 2, pp. pp. 159 –168, Feb. 2002.

[106] H. Narayanan, *Submodular functions and electrical networks*. Amsterdam:North Holland/Elsevler, 1997.

[107] V. Kumar, A. Grama, A. Gupta, and G. Karypis, *Introduction to parallel computing: design and analysis of algorithms*. Benjamin-Cummings Publishing Co., Inc. Redwood City, CA, USA, 1994.

[108] V. P. Kumar and A. Gupta, "Analyzing scalability of parallel algorithms and architectures," *Journal of parallel and distributed computing*, vol. 22, no. 3, pp. 379–391, 1994.

[109] P. Rabanal and I. Rodriguez, "Hybridizing river formation dynamics and ant colony optimization," in *Proceedings of the 10th European Conference on Advances in Artificial Life: Darwin Meets Von Neumann - Volume Part II*, ser. ECAL'09. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 424–431.

[110] C.-H. Ching and C. K. Chui, "A representation formula for harmonic functions," *Proceedings of the American Mathematical Society*, vol. 39, no. 2, pp. pp. 349–352, 1973. [Online]. Available: http://www.jstor.org/stable/2039645

[111] Y. Zhong and M. Wong, "Fast algorithms for ir drop analysis in large power grid," in *ICCAD*, 2005.

[112] B. Boghrati and S. S. Sapatnekar, "Incremental analysis of power grids using backward random walks," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 19, no. 3, p. 31, 2014.

[113] S. Nassif, "Power grid analysis benchmarks," in *Design Automation Conference, 2008. ASPDAC 2008. Asia and South Pacific*, March 2008, pp. 376–381.

[114] T. Yu and M. Wong, "Pgt solver: An efficient solver for power grid transient analysis," in *ICCAD*, 2012.

[115] J. Lampinen, "A constraint handling approach for the differential evolution algorithm," in *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*, vol. 2. IEEE, 2002, pp. 1468–1473.

[116] W. Wu, *Computational river dynamics*. Taylor & Francis London, 2008, vol. 78.

[117] M. R. Bonyadi and Z. Michalewicz, "Analysis of stability, local convergence, and transformation sensitivity of a variant of the particle swarm optimization algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 3, pp. 370–385, 2016.

[118] B. E. Hansen, "The likelihood ratio test under nonstandard conditions: testing the markov switching model of gnp," *Journal of applied Econometrics*, vol. 7, no. S1, 1992.

[119] S. Andradóttir, "A global search method for discrete stochastic optimization," *SIAM Journal on Optimization*, vol. 6, no. 2, pp. 513–530, 1996.

[120] R. Wituła and D. Słota, "Cardano's formula, square roots, chebyshev polynomials and radicals," *Journal of Mathematical Analysis and Applications*, vol. 363, no. 2, pp. 639–647, 2010.

[121] J. Liang, B. Qu, and P. Suganthan, "Problem definitions and evaluation criteria for the cec 2014 special session and competition on single objective real-parameter numerical optimization," *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore*, 2013.

[122] N. Awad, M. Ali, J. Liang, B. Qu, and P. Suganthan, "Problem definitions and evaluation criteria for the cec 2017 special session and competition on single objective bound constrained real-parameter numerical optimization," in *Technical Report*. NTU, Singapore, 2016.

[123] A. Auger and N. Hansen, "A restart cma evolution strategy with increasing population size," in *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, vol. 2. IEEE, 2005, pp. 1769–1776.

[124] S. Elsayed, N. Hamza, and R. Sarker, "Testing united multi-operator evolutionary algorithms-ii on single objective optimization problems," in *Evolutionary Computation (CEC), 2016 IEEE Congress on*. IEEE, 2016, pp. 2966–2973.

[125] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space," *IEEE transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.

[126] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y.-P. Chen, A. Auger, and S. Tiwari, "Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization," *KanGAL report*, vol. 2005005, p. 2005, 2005.

[127] K. Deb, "An efficient constraint handling method for genetic algorithms," *Computer methods in applied mechanics and engineering*, vol. 186, no. 2, pp. 311–338, 2000.

[128] A. Auger, N. Hansen, and M. Schoenauer, "Benchmarking of continuous black box optimization algorithms," 2012.

[129] M. Jamil and X.-S. Yang, "A literature survey of benchmark functions for global optimisation problems," *International Journal of Mathematical Modelling and Numerical Optimisation*, vol. 4, no. 2, pp. 150–194, 2013.

[130] B. Qu, J. Liang, Z. Wang, Q. Chen, and P. N. Suganthan, "Novel benchmark functions for continuous multimodal optimization with comparative results," *Swarm and Evolutionary Computation*, vol. 26, pp. 23–34, 2016.

[131] S. Dash, K. L. Baishnab, and G. Trivedi, "Applying river formation dynamics to analyze vlsi power grid networks," in *VLSI Design and 2016 15th International Conference on Embedded Systems (VLSID), 2016 29th International Conference on*. IEEE, 2016, pp. 258–263.

[132] T. A. Davis and E. Palamadai Natarajan, "Algorithm 907: Klu, a direct sparse solver for circuit simulation problems," *ACM Transactions on Mathematical Software (TOMS)*, vol. 37, no. 3, p. 36, 2010.

[133] X. Xiong and J. Wang, "Verifying RLC power grids with transient current constraints," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 7, pp. 1059–1071, 2013.

[134] K. Haghdad and M. Anis, "Power supply pads assignment for maximum timing yield," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 58, no. 10, pp. 697–701, 2011.

[135] S. Dash, D. Joshi, and G. Trivedi, "Cmos analog circuit optimization via river formation dynamics," in *2016 26th International Conference Radioelektronika (RADIOELEKTRONIKA)*. IEEE, 2016, pp. 51–55.

[136] Y.-T. Kao and E. Zahara, "A hybrid genetic algorithm and particle swarm optimization for multimodal functions," *Applied Soft Computing*, vol. 8, no. 2, pp. 849 – 857, 2008.

[137] T. Niknam and B. Amiri, "An efficient hybrid approach based on pso, aco and k-means for cluster analysis," *Applied Soft Computing Journal*, vol. 10, no. 1, pp. 183–197, 2010, cited By 113.

[138] X. Song, Y. Cao, and C. Chang, "A hybrid algorithm of pso and sa for solving jsp," in *Fuzzy Systems and Knowledge Discovery, 2008. FSKD '08. Fifth International Conference on*, vol. 1, Oct 2008, pp. 111–115.

[139] M. KUBAŘ and J. JAKOVENKO, "A powerful optimization tool for analog integrated circuits design," *Radioengineering*, vol. 22, no. 3, p. 921, 2013.

[140] a.H. Zaabab and M. Nakhla, "A neural network modeling approach to circuit optimization and statistical design," *IEEE Transactions on Microwave Theory and Techniques*, vol. 43, no. 6, pp. 1349–1358, 1995. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=390193

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

[142] P. E. Allen and D. R. Holberg, *CMOS analog circuit design*. Oxford Univ. Press, 2002.

[143] "http://s3.mentor.com/public_documents/datasheet/products/ic_nanometer_design/custom-ic-design/pyxis-schematic-ds.pdf."

[144] J. M. Johnson and V. Rahmat-Samii, "Genetic algorithms in engineering electromagnetics," *IEEE Antennas and Propagation Magazine*, vol. 39, no. 4, pp. 7–21, 1997.

[145] U. A. R. B. Deepak Joshi, Satyabrata Dash and G. Trivedi, "Analog circuit optimization based on hybrid particle swarm optimization," in *International Conference on Computational Science and Computational Intelligence, CSCI'15*, 2015, pp. 164–169.

[146] P. Andreani and H. Sjland, "Noise optimization of an inductively degenerated cmos low noise amplifier," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 48, no. 9, pp. 835–841, 2001.

[147] P. Andreani and H. Sjoland, "Noise optimization of an inductively degenerated cmos low noise amplifier," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 48, no. 9, pp. 835–841, 2001.

[148] J. Suzuki, "A markov chain analysis on simple genetic algorithms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 25, no. 4, pp. 655–659, Apr 1995.

[149] K. Deb, M. Mohan, and S. Mishra, "A fast multi-objective evolutionary algorithm for finding well-spread pareto-optimal solutions," *KanGAL report*, vol. 2003002, pp. 1–18, 2003.

[150] M. Iosifescu, *Finite Markov processes and their applications*. Courier Corporation, 2014.

[151] G. Rudolph, "Convergence analysis of canonical genetic algorithms," *IEEE transactions on neural networks*, vol. 5, no. 1, pp. 96–101, 1994.

[152] R. L. Tweedie, "R-theory for markov chains on a general state space i: solidarity properties and r-recurrent chains," *The Annals of Probability*, pp. 840–864, 1974.

[153] J. L. Doob, "Stochastic processes," 1990.

[154] C. M. Fonseca, L. Paquete, and M. López-Ibánez, "An improved dimension-sweep algorithm for the hypervolume indicator," in *2006 IEEE International Conference on Evolutionary Computation*. IEEE, 2006, pp. 1157–1163.

[155] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[156] C.-Y. Lee and X. Yao, "Evolutionary algorithms with adaptive lévy mutations," in *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*, vol. 1. IEEE, 2001, pp. 568–575.

[157] K. Chellapilla, "Combining mutation operators in evolutionary programming," *IEEE transactions on Evolutionary Computation*, vol. 2, no. 3, pp. 91–96, 1998.

[158] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 82–102, Jul 1999.

[159] Z. Michalewicz and C. Z. Janikow, "Handling constraints in genetic algorithms." in *ICGA*, 1991, pp. 151–157.

[160] T. Bäck and H.-P. Schwefel, "An overview of evolutionary algorithms for parameter optimization," *Evolutionary computation*, vol. 1, no. 1, pp. 1–23, 1993.

[161] K. Deb and S. Tiwari, "Omni-optimizer: A generic evolutionary algorithm for single and multi-objective optimization," *European Journal of Operational Research*, vol. 185, no. 3, pp. 1062–1087, 2008.

[162] C.-Y. Lee and X. Yao, "Evolutionary programming using mutations based on the lévy probability distribution," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 1, pp. 1–13, 2004.

[163] L. Xie and Y. Wu, "A modified multi-objective optimization based on brain storm optimization algorithm," in *International Conference in Swarm Intelligence*. Springer, 2014, pp. 328–339.

[164] X. Guo, Y. Wu, L. Xie, S. Cheng, and J. Xin, "An adaptive brain storm optimization algorithm for multiobjective optimization problems," in *International Conference in Swarm Intelligence*. Springer, 2015, pp. 365–372.

[165] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.

[166] R. R. Sarukkai, "Link prediction and path analysis using Markov chains," *Computer Networks*, vol. 33, no. 1, pp. 377–386, 2000.

[167] J. Horn, N. Nafpliotis, and D. E. Goldberg, "A niched pareto genetic algorithm for multiobjective optimization," in *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*. Ieee, 1994, pp. 82–87.

[168] "Cadence design systems, inc., virtuoso schematic editor. [online] available at https://www.cadence.com/content/dam/cadence-www/global/en_us/documents/tools/custom-ic-analog-rf-design/vir tuoso-vse-fam-ds.pdf."

[169] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm," Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, Tech. Rep. 103, 2001.

[170] A. Ghofrani, M. A. Lastras-Montaño, and K.-T. Cheng, "Toward large-scale access-transistor-free memristive crossbars," in *The 20th Asia and South Pacific Design Automation Conference*. IEEE, 2015, pp. 563–568.

[171] C. Yakopcic, T. M. Taha, G. Subramanyam, and R. E. Pino, "Generalized memristive device spice model and its application in circuit design," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 8, pp. 1201–1214, 2013.

[172] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *nature*, vol. 453, no. 7191, p. 80, 2008.

[173] J. W. Demmel, "Superlu users' guide," 1999.

[174] P. Christie and D. Stroobandt, "The interpretation and application of rent's rule," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 8, no. 6, pp. 639–648, 2000.

[175] C. Segura, A. Hernández-Aguirre, F. Luna, and E. Alba, "Improving diversity in evolutionary algorithms: New best solutions for frequency assignment," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 4, pp. 539–553, 2017.

[176] A. Trivedi, D. Srinivasan, K. Sanyal, and A. Ghosh, "A survey of multiobjective evolutionary algorithms based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 3, pp. 440–462, 2017.

[177] G. Trivedi, M. P. Desai, and H. Narayanan, "Parallelization of dc analysis through multiport decomposition," in *VLSI Design, 2007. Held jointly with 6th International Conference on Embedded Systems., 20th International Conference on*. IEEE, 2007, pp. 863–868.

[178] J. Fan, J. Wang, and M. Han, "Cooperative coevolution for large-scale optimization based on kernel fuzzy clustering and variable trust region methods," *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 4, pp. 829–839, 2014.

[179] K. Li, K. Deb, Q. Zhang, and S. Kwong, "An evolutionary many-objective optimization algorithm based on dominance and decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 5, pp. 694–716, 2015.

[180] M. Asafuddoula, T. Ray, and R. Sarker, "A decomposition-based evolutionary algorithm for many objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 3, pp. 445–460, 2015.

List of Publications

Journal Publications

1. **Satyabrata Dash**, Deepak Joshi, Ayushparth Sharma, Gaurav Trivedi, "Multiobjective Optimization using Hierarchical Nondominated Sorting Genetic Algorithm for Analog/RF Circuits", *Analog Integrated Circuits and Signal Processing*, vol. 94, pp. 27-47, 2017. DOI : https://doi.org/10.1007/s10470-017-1090-4

2. **Satyabrata Dash**, Deepak Joshi, Gaurav Trivedi, "Multiobjective Analog/RF Circuit Sizing using an Improved Brain Storm Optimization Algorithm ", *Memetic Computing*, 2018. DOI : https://doi.org/10.1007/s12293-018-0262-9

3. **Satyabrata Dash**, Sukanta Dey, Deepak Joshi, Gaurav Trivedi, "Area Minimization of Power Distribution Network using River Formation Dynamics", *Journal of Systems and Information Technology*, 2018. DOI : https://doi.org/10.1108/JSIT-10-2017-0097

4. **Satyabrata Dash**, Sukanta Dey, Anish J Augustine, Gaurav Trivedi, "Revisiting River Formation Dynamics with Improved Performance : Application in VLSI Power Distribution Network Design ", in *ACM Transactions on Design Automation of Electronic Systems*. (Under Review)

5. **Satyabrata Dash**, Gaurav Trivedi, "IR Drop Minimization in Memristor Crossbar Array using Particle Swarm Optimization with Aging Leader and Challengers", in *Neural Computing and Applications*. (Under Review)

6. **Satyabrata Dash**, Sukanta Dey, Gaurav Trivedi, "Transient Power grid Analysis using River Formation Dynamics", in *ACM Transactions on Design Automation of Electronic Systems*. (Under Review)

7. **Satyabrata Dash**, Sukanta Dey, Saumitra Sharma, M. Nikhil, Gaurav Trivedi, "A Method for Very Large Scale Optimization of Power Distribution Network using In-

telligent Decomposition Methodology ", in *IEEE Transactions on Evolutionary Computation*. (Under Review)

Conference Publications

1. **Satyabrata Dash**, Deepak Joshi, Gaurav Trivedi, "CMOS Analog Circuit Optimization via River Formation Dynamics," *in Proceedings of 26th IEEE International Conference on Radioelektronika*, Kosice, Slovakia, 2016. **(Best Paper Award)**

2. **Satyabrata Dash**, Krishna Lal Baishnab, Gaurav Trivedi, "Applying River Formation Dynamics to Analyze VLSI Power Grid Networks," *in Proceedings of 29th IEEE International Conference on VLSI Design*, Kolkata, India, 2016. **(Honorable Mention Award)**

3. **Satyabrata Dash**, Vivek Bangera, Vinay B. Y. Kumar, Gaurav Trivedi, Sachin B. Patkar, "Parallel Two Step Random Walk Algorithm to Analyze VLSI Power Grid Networks," *in Proceedings of 19th IEEE International Conference on VLSI Design and Test (VDAT)*, Ahmedabad, India, 2015.

4. **Satyabrata Dash**, Vivek Bangera, Vinay B. Y. Kumar, Sachin B. Patkar and Gaurav Trivedi, "Power grid analysis on parallel computing platforms," *in Proceedings of 25th IEEE International Conference on Radioelektronika*, Pardubice, Czech Republic, 2015.