Design Methodology for On-Chip Power Grid Interconnect: AI/ML Perspective

Sukanta Dey

Design Methodology for On-Chip Power Grid Interconnect: AI/ML Perspective

Thesis submitted in partial fullfillment of the requirements

for the award of the degree

Doctor of Philosophy

by

Sukanta Dey

Roll No.: 146201002

Under the guidance of

Prof. Sukumar Nandi

Dr. Gaurav Trivedi

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

GUWAHATI - 781 039, ASSAM, INDIA

SEPTEMBER 2020

Declaration

I, Sukanta Dey, confirm that:

- The work contained in this thesis is original and has been done by myself and the general supervision of my supervisors.
- The work has not been submitted to any other Institute for any degree or diploma.
- Whenever I have used materials (data, theoretical analysis, results) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references.
- Whenever I have quoted written materials from other sources, I have put them under quotation marks and given due credit to the sources by citing them and giving required details in the references.

Dated: 29th September 2020 Sukanta Dey

Place: Guwahati, India

Certificate

This is to certify that the thesis entitled "**Design Methodology for On-Chip Power Grid Interconnect: AI/ML Perspective**", submitted by **SUKANTA DEY** (146201002), a research scholar in the *Department of Computer Science and 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 our 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.

Dated:
Guwahati.

Prof. Sukumar Nandi
Professor
Dept. of Computer Science and Engineering
Indian Institute of Technology Guwahati
Guwahati - 781 039, Assam, India.

Dated:
Guwahati.

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

To

My ***parents*** *and* ***brother***

for their blessings, love and support

Acknowledgements

At first, I would like to thank my almighty God for their blessings throughout my life and help me achieve new heights.

This thesis would not have been possible without the immense help and support of several people in various measures, including various Internet sources. I want to convey my acknowledgment to all of them who have provided the advice and guidance to complete my doctoral degree in the Department of computer science and engineering at the Indian Institute of Technology Guwahati, India.

I wish to express my deepest gratitude to my supervisor, Prof. Sukumar Nandi, for his guidance, inspiration, and support. His timely feedback on my thesis and manuscripts have helped me to improve the quality of my work. I am also thankful to him for providing me partial financial assistance in attending conferences. Technical discussions with Prof. Nandi have always been fruitful and helped me expand my knowledge in computer science. I would also like to thank my co-supervisor, Dr. Gaurav Trivedi, for his support. Besides my supervisors, I would like to thank the rest of my thesis committee members: Prof. J. K. Deka, Prof. H. B. Nemade, Dr. John. Jose and Prof. Santosh Biswas for their insightful comments and encouragement. Further, my sincere thanks to other faculty members of the department of CSE for their constant support and help.

I am thankful to Prof. Sachin S. Sapatnekar, Professor in the Department of Electrical and Computer Engineering at the University of Minnesota, MN, USA for his precious and insightful comments on my thesis as abroad examiner. I am also thankful to Prof. Indrajit Chakrabarti, Professor in the Dept. of Electronics and Electrical Communication Engineering at IIT Kharagpur, INDIA. for his valuable inputs on my thesis as Indian examiner.

I am thankful to organizations such as ACM India-IARCS, TCVLSI - IEEE Computer Society, VLSI Society of India and the Department of CSE, IITG, for their commitment in providing travel support to attend international conferences. Also, I thank Ministry of Education (erstwhile MHRD), Government of India, for providing financial assistantship throughout the Ph.D. program.

It was a long journey of 6-7 years. Many people have helped me and motivated me throughout starting from the beginning to continue my journey of Ph.D. I am thankful to my dual degree and Ph.D. friends V A Amarnath, Abhishek, Mirza Ghalib, Rakesh Pandey, Pradeep Sharma, Rahul Gangopadhyay for their unconditional support and help during my course work and comprehensive exams,

one of the most robust period of my Ph.D. life. I would like to thank my HPC & FPGA lab seniors and friends Sushanta Bordoloi, Ashok Kumar Ray, Satyabrata Dash, Sunil Dutt, Deepak Joshi, Late Gaurav Kumar, Dheeraj Kumar Sinha, Amit Singh. Their support and motivation helped me to overcome any problems either at work and otherwise. I will cherish the moments I spent at this lab throughout my life. I would also like to thank my Network Security Lab friends, Madhurima Buragohain, Subhrendu Chattopadhyay, Pranav Kumar Singh, Pradeepkumar Bhale, where my research goals and aspirations took a proper direction. Their support and motivation helped me to grow as an individual and a researcher. Many other seniors and friends have helped me and motivated me throughout this journey of Ph.D. It would be very difficult for me to take everyone's name here. Finally, all of you made my life at IIT Guwahati a memorable experience.

I sincerely thanks to Mrs. Gauri Deori, Mr. Monojit Bhattacharjee Mr. Raktajit Pathak, Mr. Nanu Alan Kachari, Mr. Bhriguraj Borah, Mr. Pranjit Talukdar and all other department staff members for helping me different ways and at different times during my stay at IIT Guwahati.

Most importantly, none of this would have been possible without the love and patience of my family. I want to thank my parents, especially my mother and brother for being a constant source of love, concern, support, and strength all these years.

SUKANTA DEY

Abstract

Power Distribution has become a challenging issue in System-on-Chip (SoC) design. Generally, a power grid network is created for the distribution of power from the power pads to all the underlying functional blocks. During the process of power distribution, the power grid lines suffer from IR drop and electromigration (EM) issues. The IR drop occurs due to the metal resistance of the power grid lines. Due to the IR drop, functional blocks don't get proper supply voltage. If the voltage level of functional blocks goes below a certain threshold, these blocks may malfunction. Similarly, due to EM, there is a transfer of momentum from electrons to metal atoms, and movement of metal atoms happens from one site to another. Due to the movement of metal atoms, resistances of the metal lines, change and issues occur in power distribution. In some cases, movement of metals may create a discontinuity in metal lines, creating an open circuit. In other instances, metals may be deposited joining two metal lines, making short circuit connections in the power grid network, which may change the functionality of the power grid network.

Present methods of designing power grid are time-consuming as it checks for IR drop and EM violations over many iterations of the design cycle. There is a need to optimize the design cycle in order to have a fast sign-off of the power grid design stage, for future complex and intricate designs.

Therefore, in this thesis, we have provided solutions to overcome these power grid design challenges using AI/ML approaches. In our first work, we have proposed a probabilistic solution for the power grid analysis problem, with which voltages of the circuit can be obtained efficiently. We employ a Lévy flight-based jumping strategy for finding the solutions to the power grid networks. Our proposed solution is faster with the same level of accuracy than its previous works.

In our second work, design space exploration (DSE) for obtaining critical optimum power grid design is proposed using a heuristic approach. We have obtained that IR drop and metal routing area are two crucial and conflicting design objectives for power grid network. We incorporate an evolutionary computation-based heuristic to obtain the trade-off between IR drop and metal routing area of power grid network. Our proposed DSE framework helps designers get an idea about initial critical design parameters to speed up the design flow.

In our third work, we use machine learning approaches for the power grid design problem. Our objective of this work is to predict the power grid interconnect widths of all interconnects while satis-

fying certain constraints. We engage deep neural network-based machine learning techniques for the width prediction task, formulating it as a regression problem. The neural network is trained using the dataset created using standard power grid benchmarks. Our proposed framework achieves a significant speedup over the traditional approach with an acceptable accuracy limit.

In the final work, we propose an electromigration-aware lifetime of the power grid network during the design time using a machine learning technique. In this work also, we use deep neural network-based machine learning techniques for the lifetime prediction of power grid network, formulating it as a regression problem. In this case, the neural network is also trained using the dataset created using standard power grid benchmarks. Our proposed approach is faster than the state-of-the-art aging prediction models and closest to the accurate state-of-the-art physics-based model. Overall from our thesis work, we have discovered that AI/ML approaches can be a good alternative for the traditional power grid design approaches, which is fast and can speed up the overall design cycles for future intricate SoC design.

Contents

List of Figures **xix**

List of Tables **xxiii**

List of Acronyms **xxv**

1 Introduction **1**

1.1 Motivation of the Research Work 4

1.2 Contribution of the Thesis 5

1.2.1 Power Grid Analysis using Probabilistic Approach 6

1.2.2 Design Space Exploration of Power Grid using Heuristic Approach 7

1.2.3 Power Grid Design using Machine Learning 8

1.2.4 Aging Prediction of Power Grid Design using Machine Learning 9

1.3 Organization of the Thesis 10

2 Background: On-Chip Power Grid, AI/ML Approaches **11**

2.1 Introduction 12

2.2 On-Chip Power Grid Design 12

2.2.1 Overview 12

2.2.2 Power Grid Model 13

2.2.3 Power Grid Analysis 13

2.2.4 Power Grid Optimization 13

2.2.5 Power Grid Aging Prediction 14

2.2.5.1 Electromigration fundamentals 14

2.3 Review of Classical Approaches in Power Grid Design 16

2.3.1 Review on Power Grid Analysis 16

2.3.2 Review on Conventional Approaches in Power Grid Design 17

2.3.3 Review on Electromigration-aware Aging Prediction in Power Grid 18
2.4 AI/ML Techniques 19
2.4.1 Probabilistic Learning 19
2.4.2 Metaheuristic Learning 20
2.4.3 Machine Learning 20
2.5 Review of AI/ML Approaches in Power Grid Design 21
2.6 Summary 21
3 Power Grid Analysis using Probabilistic Approach 23
3.1 Introduction 24
3.2 Power Grid Model 25
3.3 Markov Chain Model With Lévy Flight 26
3.3.1 Simple Markov Chain Model 26
3.3.2 Lévy Flight 28
3.3.3 Proposed Model 29
3.4 Implementation Details 31
3.5 Experimental Results 33
3.6 Conclusion 35
4 Design Space Exploration of Power Grid using Heuristic Approach 37
4.1 Introduction 38
4.2 Preliminaries 42
4.2.1 PGN Design 42
4.2.2 PGN Model 43
4.2.3 PGN Analysis 43
4.2.4 PGN Optimization 44
4.2.5 Background of Single and Multi-objective Optimization 44
4.2.5.1 Pareto-Optimal Solution 45
4.2.6 Motivations behind using evolutionary algorithm 46
4.3 Single Objective Problem Formulation: IR Drop Minimization 46
4.3.1 Objective Function for IR Drop Minimization 46
4.3.2 Constraints for IR drop minimization 48

4.3.2.1 IR Drop Constraints . . . 48
4.3.2.2 Metal Area Constraints . . . 48
4.3.2.3 Electromigration Constraints . . . 48
4.3.2.4 Minimum Width Constraints . . . 49
4.3.2.5 KCL Constraints . . . 49
4.3.3 Minimization using Cooperative Coevolution . . . 49
4.3.4 IR drop minimization using CC-SaNSDE . . . 49
4.3.5 Experimental Results . . . 50
4.4 Single Objective Problem Formulation: Metal Routing Area Minimization . . . 51
4.4.1 Objective Function for Metal Routing Area Minimization . . . 51
4.4.2 Constraints for metal routing area minimization . . . 52
4.4.2.1 IR Drop Constraints . . . 52
4.4.2.2 Metal Line Area Constraint . . . 53
4.4.2.3 Current Density Constraint . . . 53
4.4.2.4 Metal Line Width Constraint . . . 53
4.4.2.5 Current Conservation Constraint . . . 53
4.4.2.6 Power Dissipation Constraint . . . 54
4.4.3 Metal Area minimization using CC-SaNSDE . . . 54
4.4.4 Experimental Results . . . 55
4.5 Multiobjective Problem Formulation: IR Drop and Area as Conflicting Objectives . . . 55
4.5.1 Problem Formulation . . . 55
4.5.2 Multi-objective Reliability and Yield Constraints of PGN . . . 59
4.5.2.1 IR drop Reliability Constraint: . . . 59
4.5.2.2 Metal Area-based Yield constraints: . . . 59
4.5.2.3 Electromigration Reliability constraint: . . . 59
4.5.2.4 Minimum metal line width constraints: . . . 59
4.5.2.5 Node constraints: . . . 60
4.5.3 Constraints Handling . . . 60
4.5.3.1 Boundary Constraints . . . 60
4.5.3.2 Equality Constraints . . . 60

4.5.3.3 Inequality Constraints . . . 60
4.6 Proposed Minimization Framework for PGN . . . 61
4.6.1 Minimization Framework . . . 61
4.6.1.1 IR drop analysis using KLU Solver . . . 61
4.6.1.2 Setup for Minimization . . . 62
4.6.1.3 Minimization using NSGA-II . . . 62
4.6.1.4 Selecting Appropriate Point from Optimal Pareto Front . . . 63
4.6.2 Computational Complexity . . . 64
4.7 Experimental Validation . . . 65
4.7.1 Configuration . . . 65
4.7.2 Results . . . 66
4.7.2.1 Results for a subcircuit of PGN . . . 66
4.7.2.2 Comparison of the results before and after optimization . . . 69
4.7.2.3 IR drop profile before and after optimization . . . 69
4.7.2.4 Comparison of Proposed Framework with the Simple Linear Programming Approach . . . 70
4.7.2.5 Comparative Study of Proposed Work with Our Single Objective Problem . . . 72
4.8 Conclusion . . . 73
5 Power Grid Design using Machine Learning 75
5.1 Introduction . . . 76
5.2 Preliminaries and Motivation . . . 77
5.2.1 Fundamentals of Power Planning . . . 77
5.2.2 Related Work . . . 78
5.2.2.1 Conventional Approaches in Power Grid Design . . . 78
5.2.2.2 Learning Approaches in Power Grid Design . . . 78
5.2.3 Motivation . . . 78
5.2.4 Overview of the Proposed Methodology . . . 79
5.3 Nonlinear Optimization Formulation . . . 79
5.4 Proposed PowerPlanningDL Framework . . . 81

5.4.1 Problem Formulation . . . 81
5.4.2 Feature Selection & Training Data Preparation . . . 83
5.4.3 Neural Network-based Deep Learning Model . . . 83
5.4.3.1 The Power Grid Interconnect Width Prediction . . . 84
5.4.3.2 IR Drop Prediction . . . 84
5.4.4 Test Data Generation . . . 85
5.4.5 Post-refinement Stage . . . 85
5.5 Experimental Results . . . 85
5.5.1 Simulation Setup . . . 85
5.5.2 Study of Predicted Power Grid Interconnect Width . . . 86
5.5.3 Study of Predicted IR Drop in Power Grid . . . 87
5.5.4 Main Result: Study of Convergence Time . . . 87
5.5.5 Overhead: Study of Model Accuracy . . . 89
5.5.6 Study of Variation of MSE with Perturbation Size . . . 90
5.5.7 Study of Peak Memory . . . 90
5.6 Conclusion . . . 91
6 Aging Prediction of Power Grid Design using Machine Learning 93
6.1 Introduction . . . 94
6.2 Backgrounds . . . 97
6.2.1 Electromigration fundamentals . . . 97
6.2.2 Power Grid Model . . . 98
6.3 Proposed Machine Learning Model . . . 100
6.3.1 Problem Formulation . . . 100
6.3.2 Feature Selection and Training Data Generation . . . 101
6.3.3 Proposed ML Model using Neural Network Regression . . . 103
6.3.4 Test Data Generation and Incremental Analysis . . . 106
6.4 EM Assessment using ML Model . . . 107
6.4.1 EM-Aware Aging Prediction using Proposed ML Model . . . 107
6.4.2 Proposed Failure Criterion For The PG Network . . . 108
6.4.3 MTTF Prediction . . . 111

6.4.4 Identification of EM-affected Metal Segment of PG Network 111
6.5 Experimental Results 112
6.5.1 Simulation Setup 112
6.5.2 Total Time Required for the Machine Learning Model 116
6.5.3 Results of Expected number of Mortal interconnects of PG Network 116
6.5.4 Results of MTTF Prediction and CPU Runtime 117
6.5.5 Results of Regression Model Prediction Accuracy 120
6.5.6 Comparison of MSE with Variations in Perturbation Size 123
6.5.7 Comparison of Peak Memory and CPU runtime of ML models 123
6.5.8 Results of EM-affected Metal Segment Identification 125
6.6 Conclusion 126
7 Conclusions and Future Works **129**
7.1 Introduction 130
7.2 Summary of the Contributions 130
7.3 Limitation 132
7.4 Future Works 132
A Power Grid Benchmarks **136**
A.1 IBM Power Grid Benchmarks 137
A.2 SPICE Netlist Generation 137
A.3 Sample IBM Power Grid Benchmark Netlist 138
A.4 IBM Power Grid Benchmark Statistics 142
A.5 Large Power Grid Benchmarks 143
B Metaheuristics **146**
B.1 Metaheuristics 147
B.2 Cooperative Coevolution(CC) 147
B.3 NSGA-II based evolutionary algorithm 147
Bibliography **151**
Publications Related to Thesis **157**

List of Figures

1.1 A representational view of on-chip power grid network connected with the functional blocks. These functional blocks are consist of transistors-based circuits. 2

1.2 On-Chip Power Grid Design Flow 3

2.1 Resistive network model of PGN. 14

2.2 Momentum transfer from electrons to metal atoms 15

2.3 Basic Idea of Physics-based EM model 16

3.1 (a) Power grid network model (resistive). (b) Single power grid node. [1] 26

3.2 Sample trajectory of Lévy flight of index $\beta = 1.5$ for $1K$ steps. 28

3.3 (a) Graph traversal of one instance of BDC using Lévy flight principle, (b) Corresponding BDC with transition probabilities. 31

3.4 Speedup achieved by l-BDC on GPU over other techniques (A = l-BDC_{CPU} / l-BDC_{GPU}, B = RW_{GPU} / l-BDC_{GPU}, C = CG_{GPU} / l-BDC_{GPU}) while performing on different power grid networks. 35

3.5 Percentage error evaluated for both serial l-BDC and l-BDC on GPU while performing on different power grid networks. 35

4.1 Power Planning in VLSI Physical Design with emphasis on IR drop reduction 42

4.2 A representation of floorplan of an SoC and its PGN connection with the functional logic blocks. 43

4.3 Resistive network model of PGN. 44

4.4 Decision variable and search space 45

4.5 PGN model with only one node connected to a current source. 47

4.6 A pictorial representation of metal lines of 3×3 PGN 51

4.7 PGN model with only one node connected to a current source. 58
4.8 Flow of the minimization framework 63
4.9 subcircuit of *ibmpg*1 circuit 67
4.10 Pareto front for the subcircuit showing optimal IR drop-metal routing area trade-off . 68
4.11 IR drop profile of *ibmpg*2 circuit (a) before optimization, (b) after optimization 71

5.1 Conventional Power Planning Flow in VLSI Physical Design 77
5.2 Proposed Deep Learning-based Power Planning Flow 79
5.3 Equivalence between deep neural network training and power grid design (a) Training a neural network for weights Ω (b) Solving power grid design with neural networks for weights Ω. 80
5.4 Metal line showing its features (x, y) position and I_d switching current and width w_i. . 82
5.5 (a) A floorplan of an SoC with the power grid lines over the functional blocks. (b) Variation of r^2 scores for 1000 power grid interconnects of *ibmpg*1 benchmark circuit with different input features 82
5.6 A sample neural network with three hidden layers. 83
5.7 Flow of the simulation setup of the Deep Learning Flow 86
5.8 Power Grid interconnect width prediction for *ibmpg*2 benchmark circuit (a) Correlation scatter plot (b) Error histogram (Horizontal axis represent the error). 87
5.9 IR drop map of (a) Conventional method *ibmpg*2 circuit (b) PowerPlanningDL methodology *ibmpg*2 circuit (c) Conventional method *ibmpg*6 circuit, and (d) PowerPlanningDL methodology *ibmpg*6 circuit. 88
5.10 Comparison of prediction accuracy on test set in MSE with variations in perturbations size for (a) *ibmpg*2 (b) *ibmpg*6 benchmark circuit. 90
5.11 Memory used by PowerPlanningDL for (a) *ibmpg*2 benchmark circuit and (b) *ibmpg*6 benchmark circuit. 1 Gigabyte (GB) = 953.674 Mebibyte (MiB) 92

6.1 Basic Idea of Physics-based EM model 99
6.2 (a) Floorplan of a VLSI SoC with its PG network connecting the functional blocks. (b) The resistive DC load model of the PG network. 100
6.3 Flow of Machine Learning Model for EM-aging Prediction 101

6.4 Design and Runtime parameters affecting EM MTTF [2]. 102
6.5 An illustration of neural network with its input features and output feature with magnified view of the hidden layers. 104
6.6 Nodes directly connected to the worst case IR drop noise node x 109
6.7 A PG network with 9 nodes and 12 edges 110
6.8 Flow of the simulation setup of the Machine Learning Flow 113
6.9 Expected number of mortal interconnects (η% of total PG interconnects) for different PG benchmarks until the PG network is considered dysfunctional. Here $i1$, $i2$, $i3$, $i4$, $i5$, $i6$ $in1$, and $in2$ denote $ibmpg1$, $ibmpg2$, $ibmpg3$, $ibmpg4$, $ibmpg5$, $ibmpg6$, $ibmpgnew1$, and $ibmpgnew2$ benchmark respectively. 117
6.10 Comparison of prediction accuracy on test set in MSE with variations in perturbations size for (a) $ibmpg2$ (b) $ibmpg4$ benchmark circuit. 122
6.11 Memory used by EM aging model during MTTF prediction of $ibmpg1$ benchmark circuit using (a) GPR model and (b) Neural Network model. It is to be noted 1 Gigabyte (GB) = 953.674 Mebibyte (MiB) 124

7.1 Summary of the Thesis 133

A.1 A small power grid 138

List of Tables

1.1 Summary of the contribution of the thesis and its prior works 9

3.1 Speedup Analysis of using Lévy flight Approach on CPU 34

4.1 Comparative Study of Proposed IR Drop Minimization before optimization phase . . . 50
4.2 Comparative Study of Proposed Metal Routing Area Minimization before optimization phase 55
4.3 Parameters along with their meanings 56
4.4 Parameters configuration of NSGA-II for a subcircuit 66
4.5 Data of subcircuit shown in the Figure 4.9 of *ibmpg*1 before optimization. 67
4.6 Data of subcircuit shown in the Figure 4.9 of *ibmpg*1 after optimization for the selected point (3.6e+05, 0.0878) from Pareto front. 68
4.7 IR drop before and after optimization and optimization time 69
4.8 Total Metal routing area before and after optimization 70
4.9 Comparative Study of Proposed Framework with work of single-objective formulation . 73

5.1 r^2 score of different input features and output feature w for a PG interconnect. 83
5.2 Hyperparameters used in the learning framework 86
5.3 Comparision of Worst-case IR drop using Conventional power planning approach and PowerPlanningDL framework 88
5.4 Comparison of convergence time for Conventional power planning approach and PowerPlanningDL framework 89
5.5 r^2 score, MSE and Peak memory using PowerPlanningDL framework for all the IBM PG benchmarks. Please note 1 Gigabyte = 953 Mebibyte (Mib). 90

6.1 r^2 score between MTTF vs different input features 103

6.2 Accuracy Metrics and its definition for logistic regression-based classification for detecting potentially EM-affected mortal PG interconnects. 112
6.3 Hyperparameters used in the Neural Network Architecture 114
6.4 IBM PG benchmark [3] and industry-based PG benchmarks statistics 115
6.5 Comparison of circuit analysis time for HSpice and KLU Solver 115
6.6 Training and Testing time for power grid benchmarks using Neural Network 116
6.7 Comparison of MTTF for our proposed ML-based approach with works of [4–7] for IBM power grid benchmarks. 118
6.8 Comparison of CPU Runtime for our proposed ML-based approach with works of [4–7] for IBM power grid benchmarks. 118
6.9 MSE of the EM aging models tested by different regression techniques For IBM power grid benchmarks. 121
6.10 r^2 score of the EM aging models tested by different regression techniques For IBM power grid benchmarks. 121
6.11 Peak memory used by EM aging prediction models using different regression techniques For IBM power grid benchmarks. 124
6.12 CPU Runtime of the EM aging models tested using different regression techniques For IBM power grid benchmarks. 125
6.13 Results for accuracy of the logistic regression-based EM-affected metal segment identification. 125

A.1 IBM Power Grid Benchmark Statistics for steady-state analysis 142
A.2 Large power grid benchmark for steady-state analysis 143

List of Acronyms

ANN	Artificial Neural Network
BDC	Birth Death Chain
CC	Cooperative Coevolution
DC	Direct Current
DE	Differential Evolution
DL	Deep Learning
DNN	Deep Neural Network
DSE	Design Space Exploration
EDA	Electronic Design Automation
EM	Electromigration
Fig.	Figure
GA	Genetic Algorithm
GDE	Generalized Differential Evolution
GPR	Gaussian Process Regression
IC	Integrated Circuit
KCL	Kirchoff's Current Law
KVL	Kirchoff's Voltage Law
KLU	Clark Kent LU decomposition
MiB	Mebibyte
ML	Machine Learning
MSE	Mean Square Error
MTTF	Mean-time-to-failure
NN	Neural Network
NSGA	Nondominated Sorting Genetic Algorithm

PDE	Partial Differential Equations
PDF	Probability Density Function
PDN	Power Distribution (or Delivery) Network
PG	Power Grid
PGN	Power Grid Network
PSO	Particle Swarm Optimization
ReLU	Rectified Linear Unit
SaNSDE	Self-Adaptive Differential Evolution with Neighborhood Search
SLP	Simple Linear Programming
SVM	Support Vector Machine
VLSI	Very Large Scale Integration

1

Introduction

Contents

1.1 Motivation of the Research Work 4
1.2 Contribution of the Thesis 5
1.3 Organization of the Thesis 10

The design of high-performance chips in the semiconductor industry involves a vast human workforce, which also requires a substantial amount of time for realizing the design specifications. Following the market trends, the semiconductor industry advances with new efficient technology nodes in every alternate year. Therefore, it is imperative to reduce the design cycle time so that the chip design for the present technology node can be manufactured well before the subsequent technology node evolves into the market. Otherwise, the design may become obsolete once the new technology node appears in the market. Therefore, to reduce human efforts in the design cycle and to cope with the pace needed for chip designs, automating/semi-automating, the design process is required. *Artificial Intelligence* (AI) and *Machine Learning* (ML) can play a massive role in the automation process of the chip design cycle. In this thesis, we study different roles of AI and ML in some of the crucial aspects of the chip design.

In contrast to the natural intelligence posed by humans, *artificial intelligence* is a generic term that denotes the intelligence a system or a machine poses. This intelligence can be derived by statistical learning, probabilistic learning, heuristic learning, optimization learning, learning from data, or learning from reasoning, perception, planning. Among different AI techniques, *machine learning* denotes those classes of intelligence where a system learns from its previous experiences. In this thesis, we have discussed how different learning techniques can be implemented to improve chip design tasks, especially how AI and ML can be adapted to improve the design of on-chip power grid interconnects.

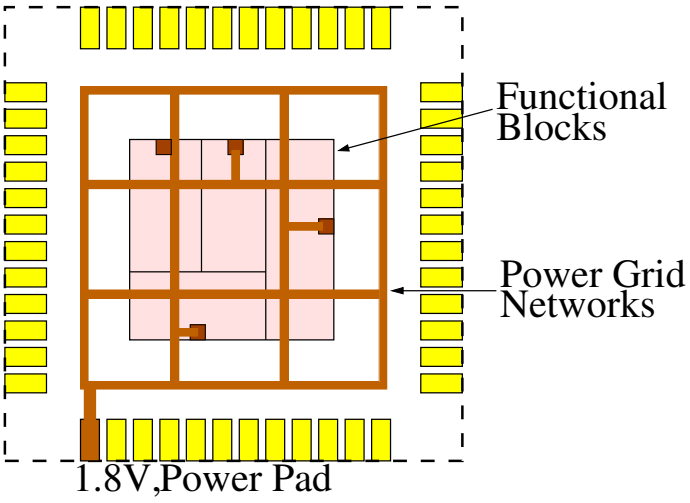


Figure 1.1: A representational view of on-chip power grid network connected with the functional blocks. These functional blocks are consist of transistors-based circuits.

On-Chip power grid interconnects are the metallic interconnects within a chip that is used to deliver

power supply voltage from the power supply pads to the underlying transistors as shown in Figure 1.1. The major challenges faced during power transfer are the following two:

- ***IR drop***: It denotes voltage drop that occurs across metallic interconnects when current is passed through it. If the IR drop exceeds a certain threshold, the underlying transistors do not get the adequate voltage that is intended by it.
- ***Electromigration***: It is the movement of metal atoms due to the exchange of momentum from the electrons to the metal atoms. A higher current density causes it. Due to the electromigration voids and hillocks are formed in the metal lines, which can short circuit or open circuit some of the metal interconnects causing malfunctioning of the chip.

If these above two challenges are not handled during the design time, the chip may malfunction, or the chip's longevity is reduced. This decreases the reliability of the chip. The conventional way of

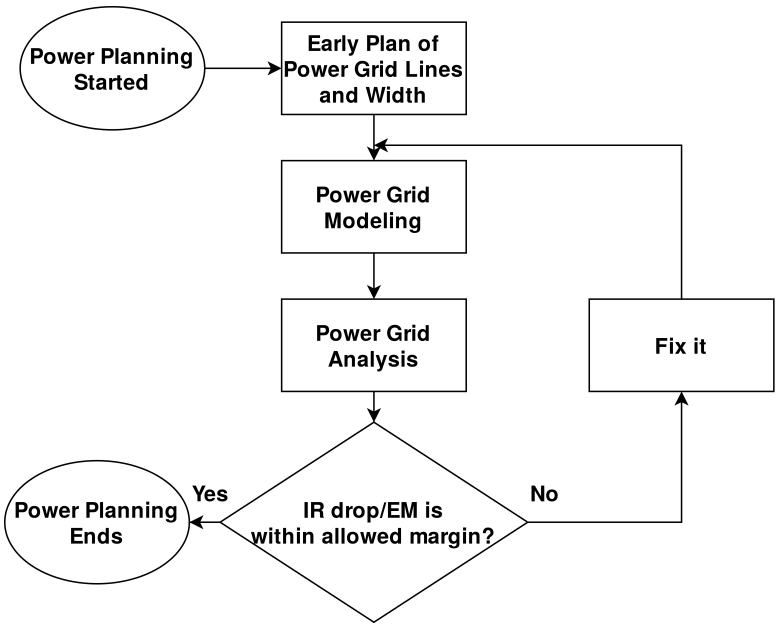


Figure 1.2: On-Chip Power Grid Design Flow

on-chip power grid design is shown in Fig. 1.2. The primary design challenge in the design of the power grid interconnects is that it takes a huge amount of circuit analysis time for a large-scale power grid network. Further, iteratively it needs to perform circuit analysis in order to optimally find the widths of the power grid interconnects. This increases the design cycle time. Also, it consumes a significant working hour. Therefore, in this thesis, we have demonstrated different AI and ML techniques to improve the power grid interconnects' design cycle.

1.1 Motivation of the Research Work

In VLSI physical design, the power planning phase is a crucial stage where power grids are designed even before the placement of underlying functional blocks or macros or intellectual property (IP) blocks, as shown in Figure 1.2. With VLSI technology's advancement, the modern-day System-on-Chips (SoC) are becoming denser with more functional blocks per unit area. Due to the scaling down of supply voltage and an increase in power demand with the integration of many functional blocks in an SoC, verifying power grids have become essential than ever to check whether all the functional blocks are getting adequate power [8]. Generally, the issues faced during the power distribution process are IR drop (voltage drop) and Electromigration related reliability issues, as discussed previously. IR drop is generated due to the inherent parasitic resistances, capacitances, and inductances of the metal lines of the power grid network. Further, the simultaneous switching of multiple functional blocks creates a significant time-varying IR drop. Moreover, electromigration deposits metal on the power grid and changes its resistance. This electromigration-induced increase in the metal resistance of the grids can also generate IR drop [9]. Consequently, due to the ill-suited design techniques, if the IR drop exceeds a certain threshold level, some functional blocks may not work correctly, which may cause malfunction of the entire SoC. Furthermore, Electromigration based reliability issues can short circuit or open circuit some portion of the metal lines over a long duration of time, which would surely decrease the lifetime of the chip.

The present methods of dealing with IR drop and Electromigration sign-off are a time-consuming process that involves many iterations in order to produce reliable power grid designs. As time progresses, designs are getting more complex and large. Present methods of IR drop and Electromigration sign-off take a considerable amount of time for larger designs and increase the design cycle time. So, it is necessary to reduce the design cycle time in order to meet the targeted design timeline for future intricate designs. Therefore, an improvement in design techniques is required for fast sign-off and reliable design of the power grid network in the power planning phase. The design cycle improvement can be incorporated if we have an efficient circuit analysis method for the power grid. It is also necessary to obtain a design trade-off of the critical parameters of the power grid to help the designers with an initial approximate design, which reduces the time for searching optimal parameters. It is also desirable to handle the Electromigration issue during the design cycle, for which it is helpful if the power grid network's accurate lifetime time can be predicted during the design time instantly.

Moreover, recent Intelligent Design of Electronic Assets (IDEA) program sponsored by DARPA in 2018 [10], focuses on creating a "no human in the loop", 24-hour turnaround layout generator for System-On-Chips, System-In-Packages, and Printed Circuit Boards. The objective of the IDEA program is to provide a path to the accelerated development of next-generation electronic systems without the requirement for a large human workforce, reducing the cost and complexity barriers associated with leading-edge electronic system design.

Henceforth, this thesis focuses on studying the issues related to large-scale power grid design in detail and presents some new approaches using AI and ML techniques. Firstly, a faster and more effective power grid analysis technique is proposed, reducing the solving time of the circuit and reducing the design cycle time. Further, it attempts to design the power grid interconnects more efficiently with AI techniques with the optimum trade-off. Furthermore, this study includes machine learning techniques for power grid design and Electromigration-aware aging prediction of the power grid network. Overall all the proposed works of the thesis help the power grid designer to obtain an initial idea of different design metrics and to handle the reliability issues in the process of designing cost-effective as well as reliable chips.

1.2 Contribution of the Thesis

This research aims to improve the power planning phase of the VLSI physical design in terms of fast design sign-off and to deal with the different reliability issues due to ill-suited design techniques. To incorporate fast sign-off, we have adopted various AI techniques, including probabilistic learning, heuristic learning, and machine learning approaches. While adopting these techniques for power grid design, reliability issues and yield are also taken care of, so that proper reliable design of power grid design can be obtained using the proposed approaches.

The significant contribution of the thesis is listed in four parts, which constitutes a chapter of the thesis each:

(i) The thesis's first contribution presents a fast power grid analysis method using the Lévy flight-based probabilistic learning principle to identify the IR drop, i.e., hotspots of a power grid network. Here, the basic idea of the proposed method is to transform the power grid network in a graph. Subsequently, traverse the power grid network following the Lévy distribution in order to find node potentials of the power grid network, from where the nodes affected from the voltage

drop noise can be located.

(ii) For the better design of the power grid, there is a need to minimize the voltage drop (IR drop) without violating the reliability constraints. Simultaneously, the yield of the design must not reduce. Therefore, in our second contribution, design space exploration is proposed using heuristic learning approaches to obtain the optimum design trade-off for the power grid interconnects.

(iii) In the third contribution, for the first time, we introduce a deep learning (DL)-based framework to approximately predict the initial design of the power grid network, considering different reliability constraints. The proposed framework reduces many iterative design steps and speeds up the total design cycle. Neural Network-based multi-target regression technique is used to create the DL model.

(iv) In the last contribution, for the first time, we propose a machine learning approach to obtain the EM-aware aging prediction of the PG network. We use neural network-based regression as our core machine learning technique to instantly predict the lifetime of a perturbed PG network.

Our proposed methods of the thesis are tested on IBM power grid benchmarks [3] and industry-based power grid benchmarks. Experimental results on these power grid benchmarks demonstrate effectiveness of the proposed approaches. Our results are also compared with state-of-the-art results, which showcase that our proposed methods are better in many scenarios for designing the power grid. More about the proposed works are mentioned below:

1.2.1 Power Grid Analysis using Probabilistic Approach

Generally, power grid network is modeled as the RLC circuit to detect the hotspots. Hotspots are the affected areas of the power grid network where the voltages level goes below a specific threshold value. Hotspots are generated due to the voltage drop across the metal lines known as IR drop. Hotspots are identified as those nodes whose voltage values drop below a specific threshold value. Therefore basic circuit analysis methods are used in the literature to detect the hotspots. However, with the increase in the size of the circuit, the traditional methods of the literature are not able to perform circuit analysis in an effective way, resulting in huge time and memory resource consumption. Therefore, researchers have used heuristic-based methods such as Random walk [11] to perform circuit analysis, which has become very famous among the research community. To make circuit analysis

more faster and effective, a method based on Levy Random Walk is used in this thesis in Chapter 3 to detect the hotspots created by the voltage drop. Circuit analysis consists of steady-state analysis and transient analysis. In this thesis, we have only considered steady-state analysis. In the steady-state analysis, only the resistive elements of the circuits are considered. From the resistive electric networks, linear system of equations are formed $GV = I$ where G, V, and I are the conductance matrices, voltage vectors and current vectors respectively, which is then solved using traditional solvers such as Gaussian Elimination, Gauss-Jordan etc. Random walk-based heuristic method is also used to solve the linear equations system in an analogous way, which has shown better performance in circuit analysis with respect to time than the traditional solvers within the acceptable limit of error. However, one demerit of it is that Random Walk depends on the random probability values to find the V_{dd} homes, so it may keep traversing into some closed loop of nodes without finding the destination home, which unnecessarily increases the solving time of power grid network. To make the circuit analysis faster, in Chapter 3, Levy Random Walk method is employed which uses jumping from one node to the other to make the traversal of the whole power grid network faster and reduces time significantly. Our proposed solution also removes the problem of trapping in a loop of nodes. To compensate for any kind of error due to the introduction of Levy Random walk, effective resistance between two points of the power grid is calculated and embedded in the equations of random walk. Finally, the performance of the proposed scheme is validated on standard power grid benchmarks, which shows significant speedup. Experimental results show that the proposed method demonstrates remarkable performance improvement over state-of-art solutions with acceptable accuracy loss ($\leq 4\%$).

1.2.2 Design Space Exploration of Power Grid using Heuristic Approach

This work deals with design space exploration of different critical power grid design objectives with the help of heuristic approaches. The design of the power grid network (PGN) of a VLSI chip is a challenging task because of increase in network complexity. Due to presence of resistances of the metal lines of the PGN, voltage drops occur in the form of IR drop, which can change the voltage level of underlying logic circuits, resulting in malfunction of the System-on-Chip (SoC). The IR drop also depends upon different reliability constraints, and violation of those constraints can deteriorate the IR drop much more. Subsequently, a significant objective while designing a PGN is to reduce the IR drop without violating the reliability constraints. IR drop also affects timing of the critical path of the circuits. Over the past two decades, several works have been proposed to optimize the PGN

by minimizing metal area considering the IR drop as a design constraint. One of the widely accepted IR drop minimization practices is by increasing the metal widths, which in turn increases the metal area. As a result, area of the chip increases, which manifests that the primary design objectives, i.e., IR drop, and metal routing area, are conflicting in nature. Therefore, these two conflicting design objectives need to be accommodated while designing the PGN in order to optimize reliability and yield of the chip. In this work, for the first time, we propose a multiobjective design space exploration framework for the power grid design, which deals with reliability and yield. We have studied various aspects of design to determine a trade- off between these two critical conflicting design objectives by developing an optimization framework using the evolutionary algorithmic technique. Results on the standard power grid benchmarks demonstrate that our proposed framework helps in a reliable PGN design with high yield. This work is described in detail in Chapter 4.

1.2.3 Power Grid Design using Machine Learning

In this proposal, we describe how the machine learning approach can be incorporated for designing reliable on-chip power grid. With increase in the complexity of chip designs, VLSI physical design has become a time-consuming task, which is an iterative design process. Power planning is that part of the floorplanning in VLSI physical design where power grid networks are designed in order to provide adequate power to all the underlying functional blocks. Power planning also requires multiple iterative steps to create the power grid network while satisfying allowed worst-case IR drop and Electromigration (EM) margin.

For the first time, this work introduces deep learning (DL)-based framework to approximately predict initial design of the power grid network, considering different reliability constraints. The proposed framework reduces many iterative design steps and speeds up the total design cycle. Neural Network-based multi-target regression technique is used to create the DL model. Feature extraction is done, and training dataset is generated from the floorplans of some of the power grid designs extracted from IBM processor. The DL model is trained using the generated dataset. The proposed DL-based framework is validated using a new set of power grid specifications (obtained by perturbing the designs used in the training phase). The results show that the predicted power grid design is closer to the original design with minimal prediction error (2%). The proposed DL-based approach also improves the design cycle time significantly with a speedup of $\sim 6\times$ for standard power grid benchmarks. From the IR drop shown in Chapter 5, it can be understood that the predicted IR drop map using the

Table 1.1: Summary of the contribution of the thesis and its prior works

Domain		Prior Art	This work
Power Grid Analysis (Chapter 3)	Steady-state analysis	Random Work, Gauss-Seidel Approach	Levy Flight-based Probabilistic technqiue
Power Grid Optimization (Chapter 4)	Design Space Exploration	Signle objective DSE (Simple Linear Programming)	Multi-objective DSE Heuristic technique
ML in Power Grid Design (Chapter 5)	Metal width prediction	Conventional approach	ML-based technique is introduced
ML in EM aging prediction (Chapter 6)	MTTF prediction	Physics-based EM model	ML-based technique is introduced

learning framework is almost similar to the IR drop map created by the conventional approach. This work is described in detail in Chapter 5.

1.2.4 Aging Prediction of Power Grid Design using Machine Learning

In this section, our proposed work depicts a machine learning approach for computing lifetime of the power grid network in its design phase. With the advancement of technology nodes, Electromigration (EM) signoff has become increasingly difficult, which requires a considerable amount of time for an incremental change in the power grid (PG) network design in a chip.The traditional Black's empirical equation and Blech's criterion are still used for EM assessment which is a time-consuming process.

In this work, for the first time, we propose a machine learning approach to obtain the EM-aware aging prediction of the PG network. We use neural network-based regression as our core machine learning technique to instantly predict the lifetime of a perturbed PG network. The performance and accuracy of the proposed model using Neural Network are compared with the well-known standard regression models. We also propose a new failure criterion based on which the EM-aging prediction is done. Potential EM-affected metal segments of the PG network are detected by using a logistic-regression based classification machine learning technique. Experiments on different standard power grid benchmarks show a significant speedup for our machine learning model compared to the state-of-the-art models. The predicted value of MTTF for different power grid benchmarks using our approach is also better than some state-of-the-art MTTF prediction models and comparable to the other accurate models. This work is described in detail in Chapter 6.

Contributions of the thesis with their prior works are listed in Table 1.1.

1.3 Organization of the Thesis

The rest of thesis is organized as follows:

Chapter 2 provides a brief discussion about state-of-art works on VLSI on-chip power grid design, including the details of different standard machine learning and optimization techniques used in the thesis.

Chapter 3 proposes a new power grid analysis method using the Lévy flight-based technique. Finally, the performance of the proposed scheme is validated on standard power grid benchmarks, which shows significant speedup.

Chapter 4 deals with the optimization of different critical power grid design objectives with the help of evolutionary computing techniques.

Chapter 5 describes the adaptation of machine learning approach in on-chip power grid design.

Chapter 6 shows a machine learning approach for computing the lifetime of the power grid network in its design phase.

Chapter 7 includes summarization of the contribution of thesis and a few possible future works.

2

Background: On-Chip Power Grid, AI/ML Approaches

Contents

2.1 Introduction 12
2.2 On-Chip Power Grid Design 12
2.3 Review of Classical Approaches in Power Grid Design 16
2.4 AI/ML Techniques 19
2.5 Review of AI/ML Approaches in Power Grid Design 21
2.6 Summary 21

2.1 Introduction

This thesis is oriented towards design of the on-chip power grid network of high performance integrated circuits. The previous chapter provided a view of challenges in the power grid design.

In this background chapter, we present a brief introduction to definitions related to the on-chip power grid design. Firstly in Section 2.2, we have described the total power grid design phase in detail. Starting from the power grid modeling, power grid analysis, power grid optimization, and power grid aging prediction. This section also includes fundamentals of electromigration and related physics-based aging models. ÂăIn Section 2.3, we review classical approaches of the power grid design. Firstly, we review the power grid analysis methods; next, we review conventional power grid design approaches. We also discuss works related to electromigration aware aging predictions. In Section 2.4, we describe about AI/ML techniques. AI/ML techniques majorly classified in probabilistic learning, metaheuristic learning, and machine learning. All these three learning approaches are also described. In Section 2.5,learning-based power grid design approaches are described.

2.2 On-Chip Power Grid Design

2.2.1 Overview

Designing reliable PGN is the main aim of the power planning phase of VLSI Physical design. Power planning is a vital step in the design of a VLSI chip. Power planning phase consists of many steps, which is explained in the Figure 1.2. Initially, the floor-planning of the PGN is done and the metal lines of the power grid are formed according to the floorplan. Thereafter, the PGN is undergone early stage power grid analysis. PGN is extracted from the real designs and approximated circuit models are used to produce SPICE netlists. Subsequently, power grid analysis (IR drop analysis) is performed to obtain an early estimation of voltages and currents of the power grid, considering current sinks as the underlying circuits. From early stage power grid analysis the hotspots created by the IR drop can be located which may suffer from IR drop violations. Those violated regions are fixed by optimizing different parameters of the PGN, especially widening metal widths around the violated regions.

2.2.2 Power Grid Model

A representation of floorplan of an SoC and its PGN connection with the functional logic blocks is shown in Figure 1.1. For our study in this thesis, a steady-state equivalent circuit model of the PGN is used, which is shown in Figure 2.1. The model includes only resistances of the metal lines ignoring all other parasitic effects and constructing a grid of resistances, Âăas shown in Figure 2.1. The functional blocks are modeled as the constant current sources connected to the ground, as shown in Figure 2.1. Similarly, for modeling the ground network, the direction of the current sources have to be reversed. All power connections (V_{dd}) should be replaced as the ground connection (GND) for the ground network. Vias used for connecting different metal layers are presumed to have zero resistance for our model, as it bears very minimal resistances. We have not consider any other effects due to the capacitances or the inductances amalgamated with the C4 bumps, which is used to connect V_{dd} and ground connections.

2.2.3 Power Grid Analysis

The main motive of performing power grid analysis (or IR drop analysis) is to find the voltage affected nodes created by the IR drop. Generally power grid analysis is done using steady state analysis and transient analysis. Initially steady state analysis is done to see the steady state current and voltages of the circuit. In this thesis, we consider only the steady state model for our problem formulation. We represent the steady-state model of the PGN as a system of linear equations i.e., $\mathbf{GV} = \mathbf{I}$, where $\mathbf{G}$ matrix denotes the conductances of the metal lines, the current sources are interpreted as $\mathbf{I}$ vector and node voltages of all the nodes of the Âăgrid form the $\mathbf{V}$ vector. We adapt a direct solver as proposed in [12] named as ÂăKLU solver, for solving the matrices and obtaining the current, voltages of the PGN. We use this current and voltages as input to our multiobjective optimization engine.

2.2.4 Power Grid Optimization

The aim of the PGN optimization is to minimize all the reliability issues especially to minimize IR drop which may occur in the power grids. Generally, after locating the IR drop affected nodes different approaches can be adopted to reduce the IR drop. In this thesis, we have adopted widening the wire length method as it doesn't require changing the topology of the grid. Accordingly we propose to automate the IR drop minimization process along with considering metal routing area as another objective and formed a multiobjective optimization problem under some reliability constraints. This

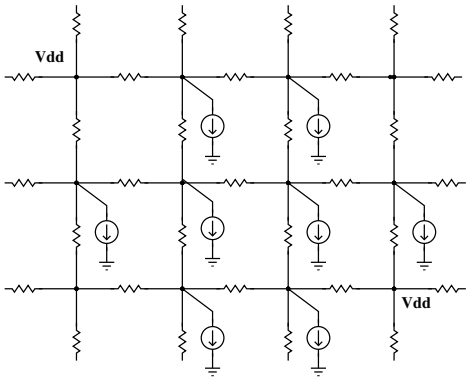


Figure 2.1: Resistive network model of PGN.

is mentioned in details in Chapter 4.

2.2.5 Power Grid Aging Prediction

It has been demonstrated that the lifetime of a chip can be affected due to deposition of metal atoms or formation of voids in the power grid line. Therefore, it is necessary to create the power grid designs in such a way so that these issues don't arise. Therefore, this effect is studied in order to obtain lifetime of the power grid network. The following subsection describes this electromigration effect in detail.

2.2.5.1 Electromigration fundamentals

Electromigration is the process of movement of metal atoms due to the exchange of momentum from the electrons to the metal atoms. The EM degradation can happen in two phases: *void nucleation* and *void growth*. Under high current in the metal lines, metal atoms are subjected to stress for a prolonged period of time, which causes void to occur. This EM degradation phase is termed as *void nucleation* phase. ÂăOnce the void nucleates, it started to grow which is termed as *void growth* phase. The aging of the metal lines due to ÂăEM degradation is measured as *mean-time-to-failure* (MTTF). ÂăBlack [13] has proposed an empirical equation to evaluate the MTTF of the metal interconnects due EM, which is given as follows,

$$MTTF = \frac{A}{J^n} e^{\frac{E_a}{kT}}, \tag{2.1}$$

which evaluates the interconnect MTTF based on known current density (J) and temperature (T). A is a constant depending on the metal geometry, grain size, and current density. The value of n is found to be 2 which, E_a is the EM activation energy, and k is the Boltzman's constant. ÂăBlech [14]

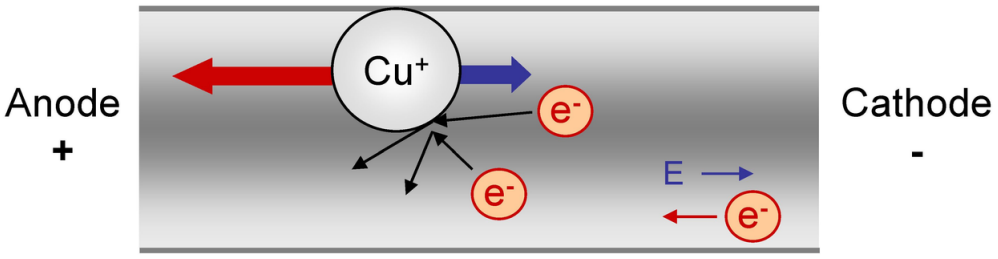


Figure 2.2: Momentum transfer from electrons to metal atoms

observed that mortality of the metal interconnects vary with the length. He proposed a criterion for the filtration of immortal interconnects, which follows,

$$(JL) \leq (JL)_c = \frac{\Omega \sigma_c}{eZ\rho} \tag{2.2}$$

L is the length of the metal interconnect, Ω is the atomic volume, e is the electron charge, eZ is the effective charge of the migrating atoms, ρ is resistivity of the metal interconnect, σ_c is the critical stress needed for failure of the metal interconnect. Korhonen et al. [15] proposed a mathematical formulation to represent the *hydrostatic stress* σ which originates from the influence of EM.

$$\frac{\partial \sigma}{\partial t} = \frac{\partial}{\partial x}\left[c\left(\frac{\partial \sigma}{\partial x} + \frac{eZ\rho J}{\Omega}\right)\right], \tag{2.3}$$

where $c = \frac{D_a B \Omega}{kT}$, where D_a is the atomic diffusivity, B is the bulk modulus. Approximate value of the nucleation time can be obtained from (2.3). A void nucleates once the stress exceeds the critical value.

Physics-based Models: The basic idea of the physics-based models is solving the interconnect trees[1], which are generated from the power grid netlist. Initially, the current densities of all the interconnect trees are calculated. Subsequently, Partial Differential Equations (PDE) associated with the interconnects (refer (2.3)) are solved in order to find the stress level. The solution of the PDE gives us the nucleation time (the time required for nucleation[2]) for critical values of stress. This nucleation

[1]Interconnect trees: connected graph of the interconnects.
[2]Nucleation: first stage of formation of a void.

time dominates the MTTF value. In [4], the authors use an iterative method that looks after the changes in stress and the power grid resistance as a function of time. With time the stress level, as well as the resistance of the power grid network, exceeds a critical value, and the simulation stops. The cumulative nucleation time is considered as MTTF of the power grid network. Power grid analysis is done in every iteration to observe the voltage drop level (V_{ir}) of the interconnects, which acts as the stopping criteria of the simulation. If the voltage drop level reaches above a threshold level (V_{th}), the power grid is considered dysfunctional, and the lifetime is calculated. This process is described pictorially in Fig. 2.3. As it is an iterative process, it takes a large amount of computational time to converge. Therefore, there is a need to have new methods for EM aging evaluation, which can converge very fast.

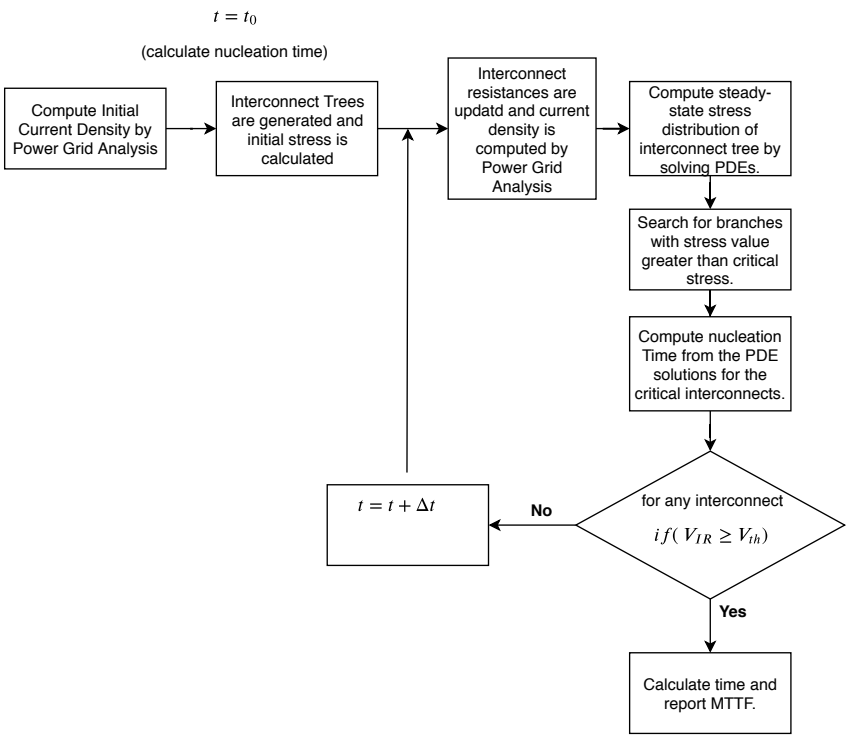


Figure 2.3: Basic Idea of Physics-based EM model

2.3 Review of Classical Approaches in Power Grid Design

2.3.1 Review on Power Grid Analysis

Analyzing worst-case IR drop by considering non-ideal power grid and underlying nonlinear circuits, is practically infeasible due to size of the power grid and also due to nonlinear nature of the many devices

in the underlying circuits. Moreover, IR drop depends on current load of the devices. Therefore, current loads are estimated before doing the IR drop analysis. Such an approximation is correct as IR drop of the PGN is generally a small percentage of the supply voltage. By modeling the underlying circuit as current loads, IR drop analysis can be formulated as a linear system of matrices. Therefore, solving this system of equations takes $\mathcal{O}(n^4)$ as system matrix is of the order of $\mathcal{O}(n^2)$. As a result, several works tried to solve the IR drop analysis using different approaches, Âăsuch as random walk [16–19] and hybrid solver [20] to do the power grid analysis without involving direct computation of the matrices related to the power grid circuit or strived to overcome size of the problem by obtaining locality of the PGN [21, 22]. IR drop analysis is done efficiently using many new approaches [23]. Some parallel approaches are also used to solve IR drop analysis [24, 25]. By doing IR drop analysis, the hotspots generated by the IR drop noise are located. Subsequently, those hotspots have to be minimized, and that is the main aim of IR drop minimization.

2.3.2 Review on Conventional Approaches in Power Grid Design

Previous work of power grid design mostly includes the IR drop limit as a design constraint in the design objective instead of minimizing it. The fundamental objective of the works in the literature is to reduce area of the metal lines of the PGN by formulating a two-phase optimization problem considering IR drop as a constraint, subsequently iteratively solve the optimization problems using different heuristics. Chowdhury's work [26], which is the first work in the literature which tried to minimize the power/ground nets area. Similarly, Chowdhury [27] tried to come up with other effective methods to optimum design for power networks. Wu et al. [28] have done the area minimization of power grid network using efficient nonlinear techniques. Mitsuhashi et al. [29] has done power and ground network topology optimization for cell-based VLSI. Tan et al. [30, 31] solved the problem of power grid optimization using a sequence of linear programming. Wang et al. [32] worked on a similar problem with the sequential network simplex algorithm. Likewise, there are several works on the metal routing area minimization. Tan et al. [33] solved the metal routing area reduction of a PGN problem using equivalent circuit modeling. Wang et al. [34] has done power ground area optimization using the multigrid-based technique. Zhuo et al. [35] also used algebraic multigrid for power grid optimization. Bhooshan [36] tried to minimize the IR drop considering the impact of resistance and inductance of the grid. Zeng et al. [37] have solved power grid optimization problem with the help of locality driven partitioning based two-step optimization algorithm. Jakushokas et al. [38] has done power network

optimization using link breaking methodology, where they demonstrated a reduction in voltage drop in the sensitive circuits. Zhou et al. [39] have done electromigration aware power grid optimization. Sadat et al. [40] have done optimal allocation of LDOs and decoupling capacitors for PGN optimization. Chakraborty et al. [41] have proposed a pre-layout and post-layout validation scheme for power grid design. Bairamkulov et al. [42] have done work on power delivery exploration methodology. However, they have done it for small power grid designs. Further, there is very few works which address the IR drop reduction with the consideration of several PGN constraints. Again, with the advancement of the FinFET based process, current density in the power grid is seen to increase by 20-30% [43], which generates additional IR drop and electromigration. Therefore, in order to increase reliability of the chip, there is a requirement of reducing this added IR drop and electromigration for the future FinFET based process designs.

There are many works in literature in the last two decades which deal with power grid designs, analysis, optimization and verification using different heuristics. Some of the recent works on the power grids are discussed here. Fawaz et al. [44] have proposed a methodology for accurate verification of the power grids. Wang et al. [45] have proposed electromigration-aware power grid design. Heo et al. [46] have done IR drop mitigation by inserting power staple. All the methods mentioned above suffer from large convergence time. Therefore, new method of power grid design is required, which can automate the power grid design stage and make the design cycle faster. In the next subsection, we discuss a review on electromigration-aware aging prediction in power grid design phase, which is also one of the important aspects of power grid design.

2.3.3 Review on Electromigration-aware Aging Prediction in Power Grid

Recently many works on physics-based EM model is proposed for optimistic aging prediction of the PG network. Huang et al. [4] have proposed such a physics-based model for the first time. Âă Mishra et al. [47] proposed a better approach for predicting lifetime of the PG network considering transient stress modeling. Chatterjee et al. [5] proposed a fast physics-based electromigration assessment using an efficient solution of linear time-invariant systems. Wang et al. [45] proposed a physics-based model using integral transformation technique. Chatterjee et al. [6] extended their work on LTI system-based EM assessment approach by incorporating macromodeling-based filtering and predictor approach. There are several other works on physics-based EM assessment model [48–50]. Practical adaptation of these physics-based methods [4–6, 45, 48–51] for a full scale EM prediction

for a chip is not possible as these methods are time-consuming. Although, recent work of Najm and Sukharev [7] shows a significant speedup for full chip simulation in EM-aging prediction by employing Monte-Carlo Simulation. However, the work of [7] still requires a fresh EM-simulation for incremental changes in the design. Therefore, in order to speed up the EM-sign off phase of the incremental PG network design, it is better to reuse the historical data (generated by the standard aging models). Âă The historical data can be utilized to create a machine learning model that can instantly predict the EM-aging of the PG network. This phenomenon also helps to facilitate a practical, fast method for full-scale EM prediction. Before using machine learning techniques for power grid design and EM aging prediction, it is necessary to know a bit more about it.

2.4 AI/ML Techniques

Artificial Intelligence (AI) and Machine Learning (ML) have gained significant attention in the last decade due to the substantial breakthrough in the deep learning models in predicting complex tasks. Another primary reason behind the success of the deep learning model is the advent of the many-core architectures (GPU), which helped the training of models in feasible time in order to predict complex tasks. However, when we talk about AI/ML, it not only means the deep learning. AI means any system that possesses intelligent decision-making capability. Further, those decision-making systems, where learning happens from its previous or historical data/experience, are known as machine learning systems. Generally, the class of AI can be broadly divided into three categories based on their type of learning. These are:

- Probabilistic Learning
- Metaheuristic Learning
- Machine Learning

2.4.1 Probabilistic Learning

In probabilistic learning, the entities perform the tasks depending on the estimated probabilities of the events. The objective is to create stochastic models that describe a series of feasible occasions. In order to do so, transition probabilities for different sets of possible events are evaluated. There are several probability models in the literature, such as Markov chain models, Queuing models, Petri nets. In Chapter 3, we use Markov chain based stochastic models to analyze the power grid analysis

problem. The power grid network is modeled as a graph and mapped as a Markov chain model. For faster traversal of the power grid network, we use Lévy flight. ÂăOur proposed approach helps in obtaining faster convergence in power grid analysis.

2.4.2 Metaheuristic Learning

In metaheuristic learning, the objective is to create mathematical cost functions. Subsequently, obtain the cost function's optimized value and its corresponding decision variables by employing heuristic search techniques. For generating the search space, many random points are generated. The cost function is evaluated in all those points of the search space. Subsequently, various search techniques are employed for efficient search. These search strategies are problem-specific. For different kinds of problems, different search strategies become suitable. There is not a single generalized search technique that can give the best result for all problems. Moreover, these search strategies can find a near-optimum solution for any complex multimodal problems. In Chapter 4, we have employed metaheuristic learning in order to obtain the optimum design space exploration of the power grid design. For that, we use cooperative coevolution, and nondominated sorting genetic algorithm (NSGA-II) approaches.

2.4.3 Machine Learning

Machine Learning mostly includes learning from data. Here the underlying philosophy is to create an inference model from the dataset. Subsequently, one is required to predict samples for a new set of specifications. Machine Learning is majorly divided into two major classes:

- Supervised Learning.
- Unsupervised Learning.

In supervised learning, all the training samples of the dataset are labeled. In unsupervised learning, the dataset is unlabelled. There are different traditional machine learning models, such as SVM, Random Forest, Gaussian Process, Artificial Neural Network. These models generally don't perform well for complex prediction tasks. Subsequently, researchers in 2010s have developed and achieved massive success with the deep neural network (DNN) having multiple hidden layers. DNN can perform prediction tasks on imagenet dataset with near-human level accuracy, which gives rise to the field of deep learning. Both supervised and unsupervised, several emerging deep learning models are coming up as the time progresses. The work of this thesis is limited only to the supervised learning models.

In supervised learning, a model is created depending on these labeled samples in order to predict the labels of new test samples. Another essential part of the machine learning model is the feature set. This thesis has used deep neural network-based machine learning models in Chapter 5 and Chapter 6. Chapter 6 also demonstrated that the deep neural network performs better than other traditional machine learning models.

2.5 Review of AI/ML Approaches in Power Grid Design

There are very less efforts for the application of learning-based methods in power grid design. However, few closely related works are discussed here. Cui et al. [52] proposed a machine learning technique for power grid analysis by doing matrix-reordering. Fang et al. [53] proposed machine learning-based dynamic IR drop prediction. Liu et al. [54] proposed power supply noise aware circuit test timing prediction using machine learning. Chang et al. [55] in their work proposed to generate routability-driven power grid network using machine learning techniques. Lin et al. [56] proposed IR drop prediction of ECO-revised using machine learning. Ye et al. [57] proposed the voltage droop mitigation using a support vector machine. Cao et al. [58] proposed a learning-based method to predict the quality of power grid network package. There is not much significant work in the literature on the deep learning-based power planning methodology.

2.6 Summary

In this chapter, we have understood the background of on-chip power grid design and related AI/ML techniques used in this thesis. From next chapter onwards, contributions are described. In Chapter 3, a fast power grid analysis technique using probabilistic technique is described.

3

Power Grid Analysis using Probabilistic Approach

Contents

3.1 Introduction 24
3.2 Power Grid Model 25
3.3 Markov Chain Model With Lévy Flight 26
3.4 Implementation Details 31
3.5 Experimental Results 33
3.6 Conclusion 35

3.1 Introduction

As VLSI manufacturing process progresses towards sub-nanometer regime, the number of transistors that can be integrated on a single chip increases along with the decrease in feature lengths and supply voltages. With increase in the number of transistors, power consumption of the system increases exponentially [59]. However, reduction in feature lengths and supply voltages result in significant variations in distribution of power across elements of the chip. Such variations in power supply may compromise the chip functionality by inducing noise in the form of potential drops (worst case voltage fluctuations, i.e. hotspots) and chip leakage currents. To estimate such instances, often extensive simulations are performed on the underlying power grid network of the chip using efficient power grid analyzers and with increase in size of the power grid network, it becomes a challenging task to run simulations without supercomputing resources.

To analyze a power grid network for hotspot estimation, the power grid can be described as a connected graph and the problem of solving this graph can be represented as a linear system of equations. Various methods have been presented in literature for analyzing power grid networks. Several direct methods are proposed to exploit sparsity and grid structure in the power grid network using different preconditioning strategies [59–62]. Although the preconditioned solver used in [59] reduces the computational complexity of steady state analysis of solving a power grid network to $O(n^2)$, it is not efficient in terms of leveraging previous simulation runs and in terms of achieving optimal convergence due to the limitations imposed by preconditioners. ÂăMethods based on hierarchical [63] and domain decomposition [60] strategies are presented to make use of the grid structure by extrapolating the results to original grid, which imposes tradeoffs between memory and runtime due to the imitations of partitioning strategies. Further, to address memory related problems efficiently, many parallel strategies [64], [65] are proposed to carry out simulations on high performance computing resources. Moreover, the cover time of many statistical techniques like random walk approach [11] takes a best case cover time of $O(nlogn)$ to the worst case of $O(n^3)$. With increase in size of the power grid networks, error in solution also increases and for complex power grid structures, convergence related issues are more prominent because of worst case performance of random walk [63]. Similar to random walk, a Markovian process can be developed to efficiently carry out simulations on large power grid

Parts of the work of Chapter 3 is published in proceedings of 30th IEEE/ACM International Conference on VLSI Design (VLSID 2017), Hyderabad, India, by S.Dey et al. "Markov Chain Model using Levy Flight for VLSI Power Grid Analysis".

networks by making modifications to process model so that convergence can be achieved with an acceptable accuracy loss.

In this chapter, a Markov chain model using Lévy flight is proposed to estimate the hotspots across large power grid networks. Power grid model is formulated as a linear programming problem for steady state power grid analysis in Section 3.2. Section 3.3 presents an overview of simple Markov chain model along with an insight into Lévy flight principle. The proposed approach is described in detail with an application to graph traversal using state transitions by introducing jump strategy of Lévy flight. Applicability of the proposed approach is shown by mapping the transition probability matrix of Markov chain to the linear programming problem of power grid analysis. Both serial and GPU implementations of proposed approach are presented in Section 3.4 to estimate the hotspots across a power grid network by making multiple transitions using Lévy flight. The improvement in runtime of using long jump strategy of Lévy flight within the Markov chain model is shown in Section 3.5 by running experiments on several benchmarks and compared with other standard power grid analyzers both in CPU and GPU platforms. It is shown that the proposed approach outperforms a direct solver, iterative solvers and random walk solver keeping the inaccuracy under a satisfactory level of 4%. Finally, section 3.6 concludes the chapter.

3.2 Power Grid Model

In this chapter, steady state power grid analysis has been performed to showcase the performance of proposed approach. To perform steady state analysis on the power grid, the edges having capacitors are made open-circuit and the edges corresponding to inductors are made short-circuit. A power grid network can be represented as a linear system of equations, i.e., $Ax = b$. In such case, the conductances contribute to formulate the coefficient matrix A and all independent sources (current sources) contribute to formulate right hand side vector b. Further, the unknown potentials at each vertices of the power grid network can be mapped to vector x and the linear system of equations can be rewritten as, $A_G V = I$, where A_G is the conductance matrix, V denotes the vector of unknown potentials and I denotes the vector of current sources of the power grid network. Nodal analysis method can be used to analyze such power grid networks, since all the voltage sources in the network are connected to the ground as shown in Figure 3.1(a). From Figure 3.1(b), it can be seen that each vertex in the power grid network is connected to its neighbors and as the power grid structure is

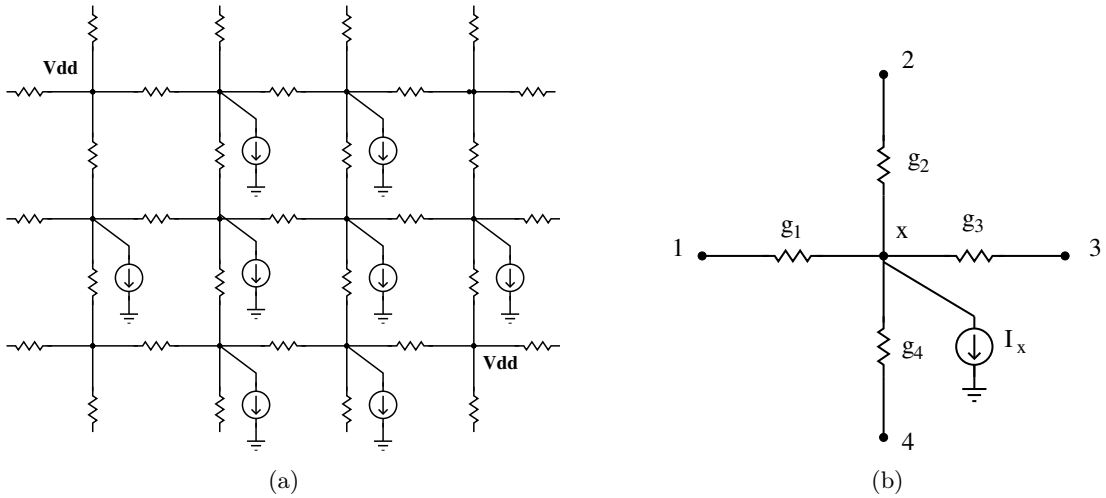


Figure 3.1: (a) Power grid network model (resistive). (b) Single power grid node. [1]

regular, the system matrix A generated using Nodal analysis is sparse, symmetric and positive definite in nature [1].

3.3 Markov Chain Model With Lévy Flight

3.3.1 Simple Markov Chain Model

Markov process is described by the change in states on a sequence of events in random systems. The Markov property to change from one state to another without influence of past on the sequence of events can be modeled to represent a directed graph D_G with vertices representing states of the system. In that case, each edge from a vertex i to vertex j can be labeled p_{ij} if $p_{ij} > 0$, where matrix $\mathbf{P}$ of the p_{ij} form the *probability transition matrix* of Markov chain model [66] and $\Sigma_j p_{ij} = 1, \forall i$. The weight $p_{ij} = \frac{1}{deg(i)}$, if j is a neighbor of i, otherwise $p_{ij} = 0$. The progress in Markov chain events can be pictured as an iterative multiplication of probability distribution vector $\hat{p}_t$ at each state by transition matrix $\mathbf{P}$, where S denotes the state space having a sequence of events. If $\hat{p}(t+1)$ denotes the probability distribution of state S_{t+1} at time $t+1$ starting with current state of S_t, then $\hat{p}(t)$ can be represented as [66],

$$\hat{p}(t+1) = \hat{p}(t)\mathbf{P} \tag{3.1}$$

where $\hat{p}(0)$ denotes the vector of probability distribution at the start vertex. In case of undirected graph U_G, Markov chain can be applied to choose any random edge incident to the current vertex uniformly and proceed towards the connected vertex at the other end of chosen edge. This property

on an undirected graph can be viewed as an analogy between Markov process and electrical network. A stochastic process on an undirected connected steady state electrical network can be modeled as a *birth-death chain* (BDC) [67] with constant birth and death probabilities. A Markov chain is said to be a birth-death chain if the transition probabilities can be defined as follows [67],

$$p_{ij} = \begin{cases} x_i, & if\ j = i+1 \\ y_i, & if\ j = i-1 \\ 1 - x_i - y_i, & if\ j = i \\ 0, & otherwise \end{cases} \tag{3.2}$$

where x_i represents the birth rate and y_i denotes the death rate. Here $x_i, y_i \geq 0$ and $x_i + y_i \leq 1$. In case of constant birth and death probabilities, $p_i = p$ and $q_i = q$, $\forall i$.

Theorem 3.1. *If D_n is an irreducible BDC with constant birth and death probabilities, then the stationary distribution associated with an initial state i is given by [67], $\pi(i) = \frac{\pi_i}{\Sigma_{j \in S} \pi_j}$, where $\pi_i = \begin{cases} 1 & if\ i = 0 \\ \frac{x_0 x_1 \ldots x_{i-1}}{y_1 y_2 \cdots y_i} & if\ i > 0 \end{cases}$*

This follows the notation that if a BDC follows a stationary distribution π at any time t, then according to principle of stationarity [66] it also maintains the same distribution at time $t+1$ and it is not imperative that the process converges to distribution π each time on U_G. Moreover, BDC process can be extended to model a connected electrical network with same stationary probability distribution π. Considering a resistive electrical network in which each edge represents the conductance g between two vertices (i, j), the associated BDC can be defined by assigning transition probability $p_{ij} = \frac{g_{ij}}{g_i}$ to edge (i, j), where $g_i = \Sigma_i\ g_{ij}$ and a stationary distribution $\pi_i = \frac{g_i}{g}$, where $g = \Sigma_i\ g_i$. This process of modeling an electrical network, and the random occurance of transitions from one vertex to another can be suitably adapted to perform steady state power grid analysis. In this case, BDC with constant birth and death probabilities can be allowed to traverse the vertices of a power grid network by following state transition events with certain transition probabilities.

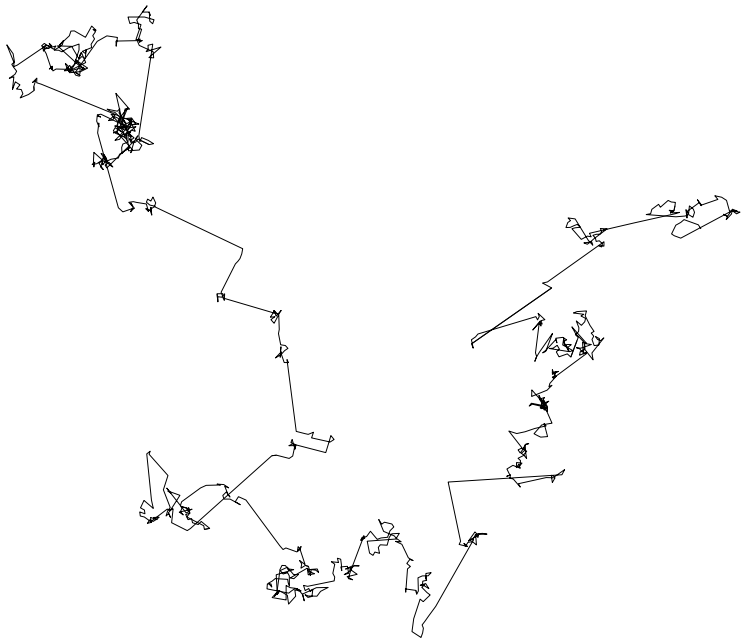


Figure 3.2: Sample trajectory of Lévy flight of index $\beta = 1.5$ for $1K$ steps.

3.3.2 Lévy Flight

Lévy flight is a special Markovian process which obeys Lévy stable distribution. Similar to Gaussian distribution that represents the limit distributions of independent identically distributed (i.i.d.) random variables with finite variance following central limit theorem, Lévy stable distribution serves as the limit distributions of i.i.d. random variables with diverging variance [68]. This favors the occurrence of long jumps in the clusters of shorter jumps as shown as an example in Figure 3.2. The jump length distribution of a Lévy flight can be approximated as a power-law as follows [66],

$$L(s) \sim s^{-1-\beta} \tag{3.3}$$

where $0 < \beta < 2$ is an index and s denotes the distance parameter [66]. However, the convergence of the associated probability density function to a Lévy stable distribution can be guaranteed with proper normalization of i.i.d. random variables to a suitable sample size, even though there is divergence across variance statistics [66]. This long-tailed jump behavior of Lévy flight can be applied to minimize the cover time of each transition states of a BDC by performing long jump events without any discontinuity in trajectory. Further, large power grid networks can be traversed with minimum computational cost by minimizing the hitting time with the benefit of probable avoidance of any self-loops during power grid analysis.

3.3.3 Proposed Model

To search for any worst-case voltage fluctuations (hotspots) across a power grid, it is necessary to analyze the whole network by progressing through individual paths covering each vertices and edges. The random probable transition from one vertex to next in a power grid network can be modeled as a state transition event in BDC with constant birth and death probabilities as described in Section 3.3.1. In view this, the model can be extended by making an addition of Lévy fight principle along the chain. In the extended model, a state in BDC can correspond to any distance neighbor states with the distance d being evaluated by Lévy flight principle. Now the state space S can be rewritten as a combination of states with $S = \{K_0, K_1, K_2, \cdots, K_d\}$, where K_i represents the state of vertex which is at i^{th} distance from current vertex.

Let x be the initial vertex of the power grid network which corresponds to K_0 state. The random probable transition from x to any vertex $x+d$ is decided by the distance parameter $0 < d < \alpha,\ (d, \alpha) \in \mathbb{N}$, where vertex $x + d$ Âăcorresponds to state K_d and α is conditional user-defined parameter. This transition process is forwarded to subsequent neighbors or any vertex of the graph uniformly at random till the stopping criteria is met. Combining all probabilities of subsequent transitions along the traversed path, the transition probability matrix $\mathbf{P}$ can be written as,

$$\mathbf{P} = \left(p_{(K_i,K_j)}\right) \quad \forall i \in \{0, n\}, \forall j \in \{0, d\} \tag{3.4}$$

where transition probability $p_{(K_i,K_j)}$ of edge (i, j) can be defined by representing Equation 3.2 as,

$$p_{(K_i,K_j)} = \begin{cases} x_i, & if\ j = i + d, d \neq 0 \\ y_i, & if\ j = i - d, d \neq 0 \\ 0, & otherwise \end{cases} \tag{3.5}$$

Further, the probability distribution of any state K_{t+n} at any time $t+n$ can be evaluated as a product of probability distribution of starting state K_t with transition probability matrix $\mathbf{P}$ by employing Equation 3.1 as,

$$\begin{aligned} \hat{p}(t + n) &= \hat{p}(t + n - 1)\mathbf{P} \\ &= \hat{p}(t)\mathbf{P}^n \end{aligned} \tag{3.6}$$

In case of power grid analysis, the transition to same state is possible for all states except the state having $d = 0$. As a power grid network can be considered as an undirected connected graph, the unknown potentials at vertices of a power grid network can be evaluated by traversing the corresponding graph and analyzing penalty associated with each vertex. The traversal along the power grid network is performed by beginning a transition from one vertex to any vertex or only neighbor vertices. The transition to any vertex from initial vertex largely depends upon parameter d evaluated using Lévy flight principle. This may give rise to two possible ways to traverse a power grid network.

Case 1 : *Transition to neighbor vertices only.*

Case 2 ($1 \leq d \leq \alpha$): *Transition to any vertex.*

In first case, BDC has only two states, i.e. K_0 state and K_1 state. The transition happens from K_0 state to K_1 state and the traversal occurs from initial vertex to any connected Âăneighbors uniformly at random. However, in second case, the transition is going to take place by a jump from one vertex to any vertex of the graph and the BDC has multiple transitions between different states depending upon the parameter d derived from Lévy flight principle. To showcase the applicability of second case, let us consider a graph having 16 vertices as shown in Figure 3.3(a), where the Lévy flight starts from vertex 1 and ends at vertex 13. The corresponding BDC model can be derived by evaluating the parameter d at each state as shown in Figure 3.3(b). Here, there are two long jumps from vertex 2 to vertex 7 with $d = 5$ (K_5 state) and vertex 7 to vertex 11 with $d = 4$ (K_4 state). The objective of this jumping scheme of graph traversal is to reduce the cover time to reach destination.

Here, for implementation of BDC model to evaluate the unknown potential of vertices of a power grid network, a V_{dd} supply power grid is taken as shown in Figure 3.1(a). A power grid network can be regarded as an undirected connected graph, which can be analyzed by representing the network in the form of linear system of equations ($A_G V = I$). One of the most probable way of analyzing the power grid network is to allow the BDC model with constant birth and death probabilities using Lévy flight principle to pursue different random paths to search for vertices having known potentials, i.e., V_{dd}. Similar to the graph traversal as shown in Figure 3.3(a), a flight can be set to follow random paths along the power grid network with long-tailed jumps depending upon parameter d. In view of this, the transition probability matrix of the power grid network (P_g) can be formulated as, $P_g \approx GA_G$, where G is the diagonal matrix and A_G represents the adjacency matrix of power grid network representing the system of equations. Moreover, multiple instances of flights can be employed to follow different

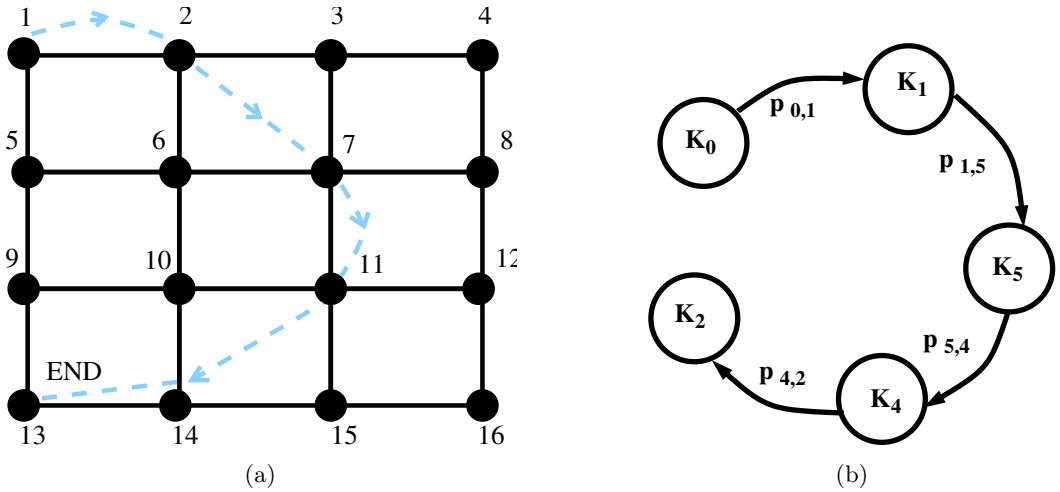


Figure 3.3: (a) Graph traversal of one instance of BDC using Lévy flight principle, (b) Corresponding BDC with transition probabilities.

random probabilistic paths from origin across a power grid network to reach V_{dd} vertices. In such cases, the flight having the least cover time is taken as the best one, i.e. shortest path to reach destination.

3.4 Implementation Details

This section depicts the framework of proposed BDC model (with constant birth and death probabilities) using Lévy flight principle to perform steady state power grid analysis by solving the underlying linear system of equations. The proposed model begins the process by initiating different instances of flights to set paths across power grid network edges. The random search for an edge at each cross-section of a power grid network largely depends upon the transition probabilities between connected vertices (for neighbor vertices only). After transition, potential value of the vertex is updated by evaluating the transition penalty associated with that vertex. The transition penalty is Âă computed by evaluating the potential across the current sink branch of that vertex using Equation 3.7 as [11],

$$p_i^{penalty} = -\frac{I_i^s}{g_i} \tag{3.7}$$

where $g_i = \Sigma_j g_{ij}$ and I_i^s represents the amount of current sinking into vertex i. For a uniform regular power grid network having equal conductances, $g_i = \frac{1}{deg(i)}$, i.e. reciprocal of degree of vertex. However, if the model follows Lévy flight, it is not imperative that all transitions happen to the neighbor branches of the power grid network and alternately it results in long-tailed jumps without any discontinuity in trajectory. For a long-tailed jump across the power grid network, the transition penalty can be

evaluated by rearranging Equation 3.7 as follows.

$$p_i^{penalty} = -\frac{I_i^s}{g_i + \delta g_{ij}} \tag{3.8}$$

$$where\ \delta g_{ij} = \begin{cases} 0, & if\ neighbor\ (i,j) \\ w, & else \end{cases}$$

Here δg_{ij} represents the weight assigned to the flight jump from vertex i to vertex j. It is set to zero if there is a transition to neighbor vertex and in all other cases, value is set to $0 < w < R_{eff}$, where R_{eff} is the effective resistance of the metal lines between the jumping points. R_{eff} can be evaluated using the analytical expression given in [69]. The process of long jumps is continued till it encounters a vertex having known potential, i.e. V_{dd}. Once the process reaches V_{dd}, the potential of the start vertex is evaluated by combining all estimated transition penalties along the traversed path. This process of estimating the potential across different vertices of power grid network comes by shear approximation to the closest of millivolts with small inaccuracy. This size of inaccuracy is measured for each solution by computing $L2\text{-}norm$ [1]. This inaccuracy in the solution can hardly be neglected if the BDC traversal process induces revisits to the current node after one step and falls into a loop,i.e. local trap. Such traps are so frequent in case of large complex power grid networks, that it can significantly deteriorate the overall performance of BDC process. To avoid such cases, BDC process is optimized by employing jump strategy of Lévy flight principle by minimizing multiple revisits to already covered vertices. The Lévy flight-based jump strategy also helps in removing the self-loops which occur in the earlier work of [11]. As a result, the walk reaches its destination quickly and we achieve a fast convergence of the power grid analysis problem. Therefore, our proposed method is faster and we have achieved a significant speedup over the [11].

Further, to accelerate the convergence of proposed Lévy flight based BDC approach to estimate hotspots across large power grid networks, the proposed model is implemented on GPU platform. As large power grid networks impose a great demand on high performance computation and memory, it is necessary to make use of powerful accelerators, like GPUs using CUDA programming model [70] that can employ kernels effectively to improve performance of proposed approach. Here a number of threads are launched in parallel to execute the proposed process on this massively SIMD architecture.

Although the number of threads is specified at the beginning of execution, the proposed process is allowed to make use of each individual thread organized in an array of a block inside kernel function while traversal. The kernel function is started by initialization of block size ($blockDim$) and grid size ($gridDim$) variables and ended by setting the total number of vertices in the power grid network to $\frac{totalNode * blockIdx+1}{gridDim}$, where $blockIdx$ represents the block index value. During kernel function implementation of the proposed process, the grid size is set to 1000 with a block size of 200 for running threads effectively on the same block. However, there is hardly any use of helper kernels to effectively maintain the synchronization among threads of different blocks and only simple kernels are executed to showcase a performance improvement of 2.48× over CPU implementation of Lévy flight based BDC approach on a power grid network having 49 million vertices.

3.5 Experimental Results

Various experiments are performed on different power grid benchmarks to showcase the effectiveness of proposed BDC model using Lévy flight approach (l-BDC). All algorithms are implemented in C/C++ using Linux environment on a machine having Intel Xeon E5-2650 processor with a connected Nvidia Tesla K20c GPU (approx. 2500 CUDA cores, 5GB global device memory). All necessary environment variables are adjusted to account for any interprocess communication during CPU-GPU transfers using $cudaMemcpy$ command [70]. The benchmarks are generated using our in-house power grid planner without the loss of generality as reported in literature [1]. The regular topology of power grid network is selected for benchmark generation in SPICE format with a grid size range of 10 thousand to 49 million vertices. The resistance values along the metal stripes are set to 0.1Ω like real industrial designs. The PADs are placed randomly across the power grids following similar industrial designs. All PADs are connected to a common power line having $1.8V$ potential and current sinks of $1mA$ are distributed across the power grid reasonably and randomly except at the PAD places.

A sequential version of l-BDC is implemented on several benchmarks to perform steady state power grid analysis to estimate hotspots and the solutions are compared with the solutions obtained from serial random walk (RW) [11], a hybrid linear solver (HLS) [71] and Gauss-seidel (GS) solver to showcase the improvement in terms of runtime. As listed in Table 3.1, speedups of 60.39× and 10.43× has been achieved over serial RW solver and HLS respectively for a power grid network having 49 million vertices. Both solvers are standard power grid analyzers reported in literature. The reason for

Table 3.1: Speedup Analysis of using Lévy flight Approach on CPU

Nodes Âă	t_{RW} (s)	t_{GS} (s)	t_{HLS} (s)	t_{levy} (s)	Speedup (t_{RW}/t_{levy})	Speedup (t_{GS}/t_{levy})	Speedup (t_{HLS}/t_{levy})
pgckt_10K Âă	0.06	0.07	0.22	0.04	1.50×	1.75×	5.50×
pgckt_40K Âă	0.30	0.36	0.82	0.17	1.76×	2.11×	4.82×
pgckt_90K Âă	0.65	1.62	1.84	0.30	2.16×	5.40×	6.13×
pgckt_250K Âă	1.92	6.78	6.06	0.86	2.23×	7.88×	7.04×
pgckt_640K Âă	7.98	19.31	20.00	2.19	3.64×	8.81×	9.13×
pgckt_1M Âă	18.95	27.85	39.55	3.51	5.39×	7.93×	11.26×
pgckt_4M	297.21	117.76 Âă	154.78	14.61	20.34×	8.06×	10.59×
pgckt_9M Âă	1513.4	272.74 Âă	349.66	33.88	44.66×	8.05×	10.32×
pgckt_16M Âă	3326.44	486.03 Âă	651.15	61.49	54.09×	7.90×	10.58×
pgckt_25M Âă	6263.80	760.10	1034.36	112.85	55.50×	6.73×	9.16×
pgckt_36M Âă	9800.35	1094.01	1562.71	167.63	58.46×	6.52×	9.32×
pgckt_49M Âă	14065.90	1498.20	2430.38	232.91	60.39×	6.43×	10.43×

this improvement in runtime is because of fewer revisits to already covered vertices across the power grid network and the effective use of jump strategy of Lévy flight principle. Further, the solutions of l-BDC are compared with the solutions of GS solver, and speedups of 6.43× have been achieved over GS (for $pgckt\_49M$). Improvement over GS solver is due to slow rate of convergence of the iterative solver. On the other hand, the proposed approach does not engage in time consuming computations involving matrix factorization or formulation.

Further, to improve the performance of l-BDC over large power grid networks ($pgckt\_1M$ – $pgckt\_49M$), extensive experiments have been carried out on GPU and a speedup of 2.48× has been achieved over its serial counterpart (for $pgckt\_49M$). The speedup can further be improved by executing smarter kernels, such as $timedevX$ series, $14dx\_s$ kernel etc. and by employing better synchronization methods among threads of different blocks. However, as compared to simple kernels, implementing complicated kernels is not an easy task. Complicated kernels make use of numerous memory access operations and kernel registers to speed up the system performance. However, it comes at the cost of register spills. In such cases, if by any means the register spills can be minimized by making improvements to the program, the kernel speed is capitalized significantly. In addition, we have also compared the solutions of l-BDC with the solutions obtained by implementation of GPU versions of random walk (RW_{GPU}) solver and an efficient iterative conjugate gradient (CG_{GPU}) solver [59] and it can be seen from Figure 3.4 that speedups of 1.99× and 46.09× has been achieved over RW_{GPU} and CG_{GPU} respectively on $pgckt\_49M$.

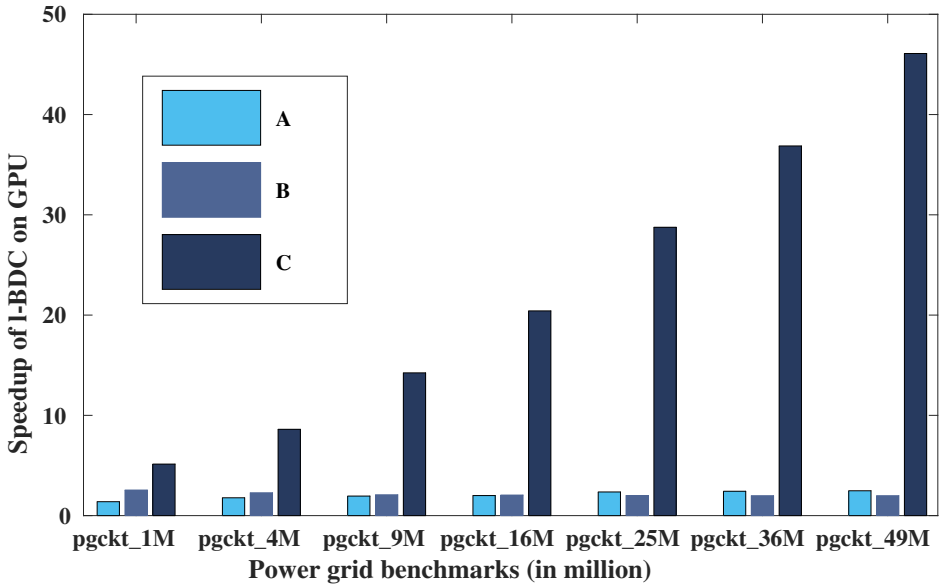


Figure 3.4: Speedup achieved by l-BDC on GPU over other techniques (A = $l\text{-}BDC_{CPU}$ / $l\text{-}BDC_{GPU}$, B = RW_{GPU} / $l\text{-}BDC_{GPU}$, C = CG_{GPU} / $l\text{-}BDC_{GPU}$) while performing on different power grid networks.

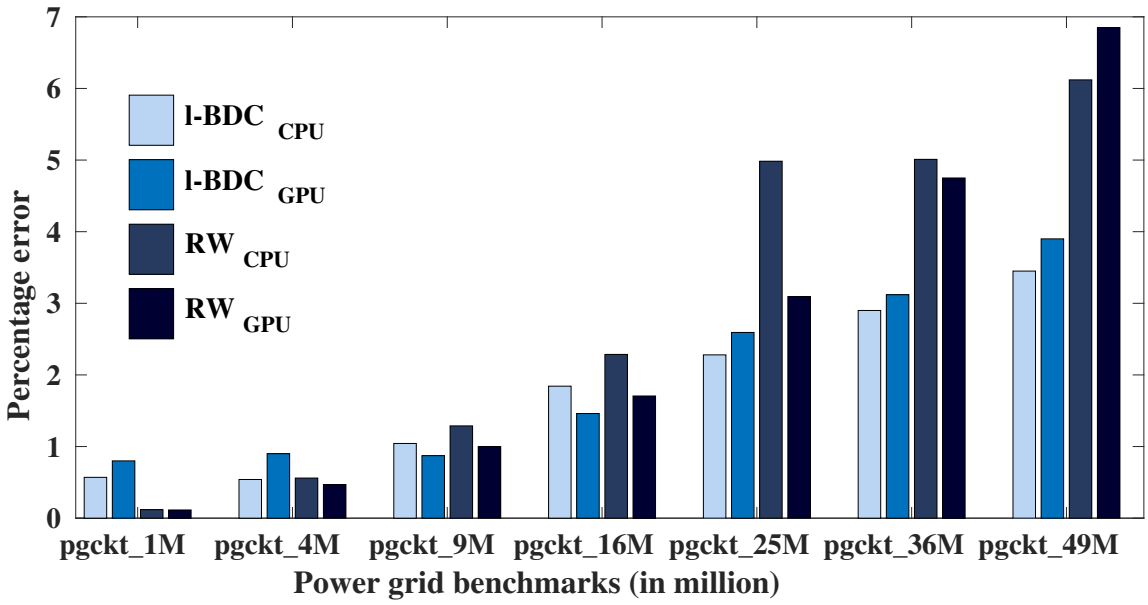


Figure 3.5: Percentage error evaluated for both serial l-BDC and l-BDC on GPU while performing on different power grid networks.

3.6 Conclusion

This chapter presents a power grid analyzer based on Lévy flight principle. This chapter presents a probabilistic method of solving power grid analysis problem based on Lévy flight principle. In general, power grid analysis problem can be formulated as a system of linear equations. We present an equivalent

probabilistic approach, which can be employed to obtain the power grid network's solutions in a fast way. In this work, Lévy flight approach is used to traverse the power grid network, which helps obtain a fast solution of the power grid network. We also remove the self-loops in our proposed approach, which also helps in fast convergence in power grid network solutions. Our proposed approach is validated using large-scale power grid benchmarks. Results show significant speedup over the Random walk and Gauss-Seidel approach. Both serial and GPU implementation of the proposed approach is presented to showcase the efficiency in runtime.

In this chapter, we have observed a fast power grid analysis method using probabilistic technique. Power grid analysis is a part of the total power grid design cycle. In order to design the power grid, we need to optimize different critical design objectives which is described in the next chapter.

4

Design Space Exploration of Power Grid using Heuristic Approach

Contents

4.1 Introduction 38
4.2 Preliminaries 42
4.3 Single Objective Problem Formulation: IR Drop Minimization 46
4.4 Single Objective Problem Formulation: Metal Routing Area Minimization 51
4.5 Multiobjective Problem Formulation: IR Drop and Area as Conflicting Objectives 55
4.6 Proposed Minimization Framework for PGN 61
4.7 Experimental Validation 65
4.8 Conclusion 73

4.1 Introduction

VLSI chip manufacturing is a time-taking process which includes the important power planning phase where design, analysis, and optimization of the on-chip power grid network (PGN) is done to achieve the reliable chip with high yield [72, 73]. With the advancement of technology node, power planning phase becomes very challenging. The primary objective of the power planning phase is to ensure that all the on-chip components have sufficient power and ground connections so that all the chip components work properly with adequate voltage level. However, due to existence of the resistance of metal lines of the PGNs some undesirable IR drop occurs which reduces voltage level of many on-chip components. IR drop problem can be solved by having smaller current or lower metal resistance, however it becomes challenging due to the scaled-down metal pitch and an increase in power consumption. Also, lower supply voltage has decreased the tolerance of IR drop, and chips may breakdown if the IR drop exceeds the tolerance level. Therefore, there is a need to reduce unnecessary IR drop of the PGN. Several approaches such as 1) Widen metal lines 2) Add decoupling capacitors to the design. 3) Add more V_{dd} pads to the design etc. [74] are used in the industry by the designers who manually fix the IR drop problem by changing the design of circuit and layout iteratively till the IR drop problem is within the acceptable tolerance level. Increasing the width of metal lines is one of the primary ways of reducing IR drop widely accepted in the industry [74], as this method do not attempt to change topology of the circuit or the layout. Also, widening metal width decreases the metal resistance at the expense of metal routing area which in turn reduces the IR drop across the metal layer. This approach of IR drop reduction technique can be adapted to establish an automated environment to get a budget of the metal widths within acceptable IR drop limit. However, this approach increases the metal routing area of the chip. It is also desirable for the power grid designers to have a minimum area of metal wires of the PGN in order to obtain a lesser chip area and greater yield of the chip. Therefore, these two power grid design objectives are violating such that reducing IR drop increases the area of metal wires of the power network and vice versa. Therefore, it is necessary to have a trade-off between these two objectives, i.e., IR drop and metal routing area resource of the chip in an automated environment, which is also the main motivation of this work. So minimizing these two parameters of the on-chip power network concurrently constitute a multi-objective on-chip

Parts of the work of Chapter 4 is published in proceedings of IEEE Computer Society Annual Symposium on VLSI (ISVLSI 2018), Hong Kong SAR, China, by S.Dey et al. "PGIREM: Reliability-Constrained IR Drop Minimization and Electromigration Assessment of VLSI Power Grid Networks using Cooperative Coevolution".

IR drop-metal routing area minimization problem of PGN.

Analyzing worst case IR drop by considering non-ideal power grid and underlying nonlinear circuits, is practically infeasible due to size of the power grid and also due to the nonlinear nature of many devices in the underlying circuits. Moreover, IR drop depends on the current load of the devices. Therefore, current loads are estimated before doing the IR drop analysis. Such an approximation is correct as the IR drop of the PGN is generally a small percentage of the supply voltage. By modeling the underlying circuit as current loads, the IR drop analysis can be formulated as a linear system of matrices. Therefore, solving this system of equations take $\mathcal{O}(n^4)$ as the system matrix is of the order of $\mathcal{O}(n^2)$. So, several works tried to solve the IR drop analysis using different approaches, such as random walk [16–19] and hybrid solver [20] to do the power grid analysis without involving direct computation of the matrices related to power grid circuit or strive to overcome size of the problem by obtaining locality of the PGN [21, 22]. IR drop analysis is done efficiently using many new approaches [23]. Some parallel approaches are also used to solve IR drop analysis [24, 25]. By doing IR drop analysis, the hotspots generated by the IR drop noise are located. Subsequently, those hotspots have to be minimized, and that is the main aim of IR drop minimization.

Motivations: Most of the works on power grid design optimization as discussed earlier have emphasized to optimize a single design objective of the PGN using IR drop as a design constraint. For the older technology nodes, power grid design with a single design objective is successful. However, with the advancement in technology nodes and an increase in the size of PGNs, the IR drop problem is deteriorating much more due to the increase in power consumption by the underlying circuits [43]. Simultaneously, it is also desirable to have a minimum area of the power grid. Therefore, it is more robust if we consider multiple design objectives during the design of the PGN. Furthermore, no work has been reported yet to deal with different design objectives of the power grid design problem. Minimization of only metal routing area of the PGN leads to a decrease in width of the metal wires, which in turn increases the resistance of the metal wires. As a result, the IR drop caused by the metal lines increase. Considering this if we try to minimize only the IR drop by increasing width of the metal wires then area of the metal wires increase. Hence there must be a trade-off between metal routing area resource and IR drop noise to have an optimum reliable design of the PGN. Accordingly, an optimization problem can be formed considering these two violating objectives of the power grid design.

It has been observed that variations of IR drop increase with an increase in size of the PGN. One major reason behind increase in the IR drop with increase in size of the PGN is due to an inadequate increase in the number of voltage pads (due to the area constraints) with respect to the increase in the size of PGN. Therefore, it is necessary to increase number of voltage pads in order to reduce the IR drop, which also increases area of the chip. Further, it adds other issues associated with the voltage pads (C4 pads) such as inductive noise, packaging effect on Cu/low k interconnect, and electromigration of solder balls [75, 76] which increase the reliability issues of the chip. However, reducing IR drop by widening metal width doesn't have any such reliability issues. Although it increases the metal routing area, ultimately, it results in substantial reliability in design. Furthermore, with an increase in size of the PGN, the metal routing area also increases; simultaneously, the IR drop noise deteriorates considerably for larger power grids. Moreover, minimizing only the IR drop at the expense of area is not cost-effective. Therefore, obtaining a trade-off between IR drop-metal routing area is very much required to obtain a reliable chip with acceptable IR drop also with a good yield of the chip. Therefore, in this chapter, we solve this multi-objective optimization problem with the help of metaheuristics. We have developed a framework to solve the multi-objective on-chip PGN problem. NSGA-II [77] is used as the main optimizer. The reason behind the use of NSGA-II is it deals with multi-objective conflicting optimization problem-solving. The proposed framework employs NSGA-II based optimization engine to obtain a trade-off between the total IR drop of the whole PGN and the metal routing area by varying metal line widths of each of the metal segments (branches). This facilitates the power grid designers to obtain the IR drop and metal routing area trade-off without varying the designs iteratively. Designers also utilize decoupling capacitances at a few crucial nodes to decrease the effects of IR drop and Ldi/dt voltage. However, in this chapter, we are only examining IR drop reduction by modifying the metal widths.

In this work, we are the first to:

- Propose a design space exploration framework for multobjective large-scale on-chip power grid design.
- We solve this multiobjective optimization problem using evolutionary computing technique.

With this work, we answer the following questions at VLSI Physical Design level:

- How effective evolutionary computing technique is for the on-chip power grid design phase?

- How effectively evolutionary computing technique can handle multiple design objectives in on-chip power grid design?

Initially, we have addressed the issues of IR drop and metal routing area minimization separately as single-objective optimization problems. In Section 4.3, IR drop minimization problem for PGN is addressed and is also formulated as a large-scale minimization problem. The minimization problem is solved using Cooperative Coevolution based method. Similarly, in Section 4.4, minimization of the area of the PGN is performed considering different reliability, and power-aware constraints. The primary contribution this chapter is single-objective and multi-objective optimization of power grid network. The major contributions of this chapter are listed below:

- IR drop minimization problem is addressed formulating it as objective function and solved using Cooperative Coevolution technique.
- Further, reliability-constrained problems are identified for multiple critical design objectives which is essential for addressing key challenges of reliable early stage PGN design.
- First-of-a-kind multi-objective minimization for PGN is introduced, and the objective function for the minimization of IR drop-metal routing area is formulated.
- NSGA-II based evolutionary algorithm has been adapted for minimizing the multi-objective minimization of IR drop-metal routing area.
- Experimental validations are shown for real design based power grid benchmarks, which shows an excellent trade-off between IR drop and area of the PGN, for a reliable design of PGN.

The rest of the chapter is organized as follows. The challenges and opportunities in early stage power grid design are identified and the related works are discussed in Section 4.2. IR drop minimization as single objective optimization problem is addressed in Section 4.3. Metal routing area minimization as single objective optimization problem is addressed in Section 4.4. Further, IR drop and metal routing area are formulated as multi-objective optimization problem in Section 4.5. NSGA-II based multi-objective optimization is described and adapted to minimize IR drop-metal routing area in Section 4.6. Experimental validations on different real design based power grid benchmark is shown in Section 4.7. The work is concluded in Section 4.8.

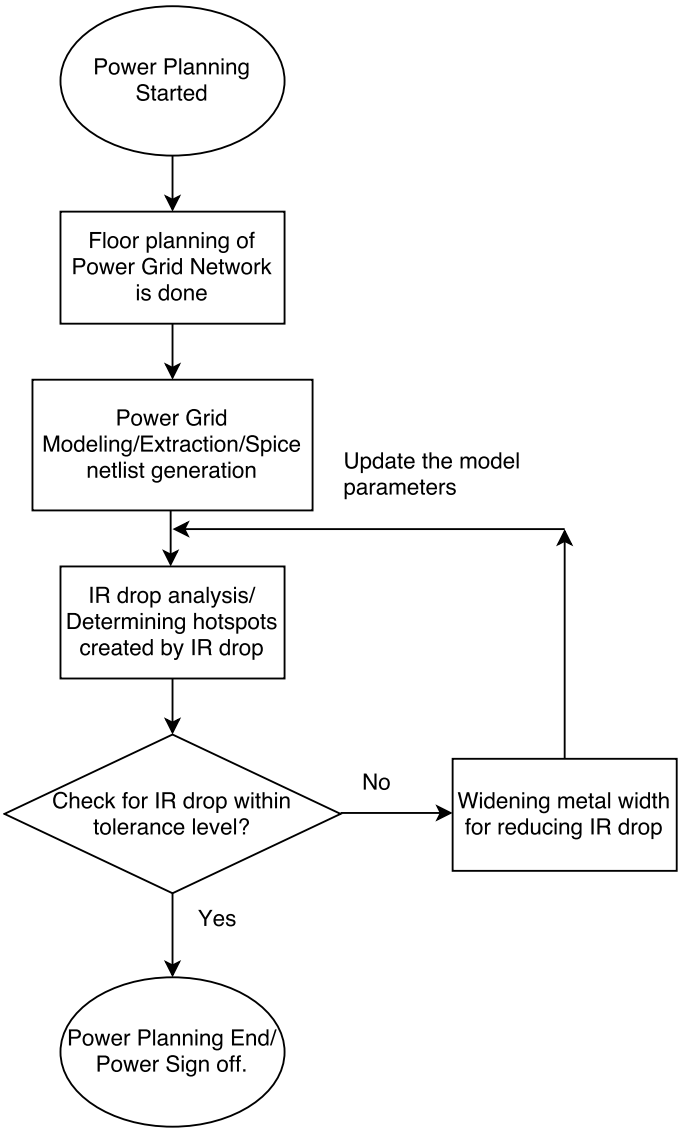


Figure 4.1: Power Planning in VLSI Physical Design with emphasis on IR drop reduction

4.2 Preliminaries

4.2.1 PGN Design

Designing reliable PGN is the main objective of the power planning phase of VLSI Physical design. Power planning is a vital step in the design of a VLSI chip. Power planning phase consists of many steps, which is explained in the Figure 4.1. Initially, the floor-planning of the PGN is performed. Subsequently, metal lines of the power grid are formed according to the floorplan. Thereafter, the PGN is undergone early stage power grid analysis. PGN is extracted from the real designs and approximated circuit models are used to produce SPICE netlists. Accordingly, power grid analysis (IR

drop analysis) is performed to obtain an early estimation of voltages and currents of the power grid, considering current sinks as the underlying circuits. From early stage power grid analysis the hotspots created by the IR drop can be located which may suffer from IR drop violations. Subsequently, those violated regions are fixed by optimizing different parameters of the PGN, especially widening metal widths around the violated regions.

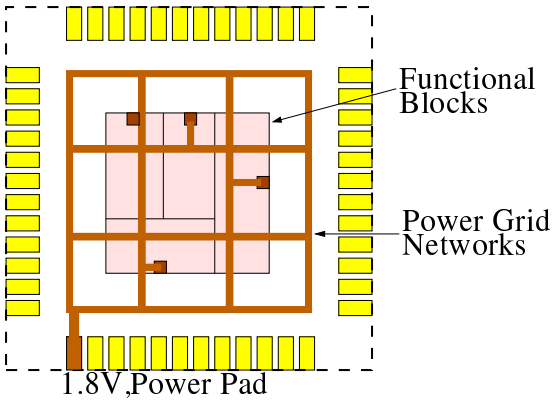


Figure 4.2: A representation of floorplan of an SoC and its PGN connection with the functional logic blocks.

4.2.2 PGN Model

A representation of floorplan of an SoC and its PGN connection with the functional logic blocks is shown in Figure 4.2. For our study, a steady-state equivalent circuit model of the PGN is used, which is shown in Figure 4.3. The model includes only the resistances of the metal lines ignoring all other parasitic effects and construct a grid of resistances, as shown in Figure 4.3. The functional blocks are modeled as the constant current sources connected to the ground, as shown in Figure 4.3. Similarly, for modeling the ground network, the direction of the current sources have to be reversed. All power connections (V_{dd}) should be replaced as the ground connection (GND) for the ground network. Vias used for connecting different metal layers are presumed to have zero resistance for our model, as it bears very minimal resistances. We have not considerd any other effects due to the capacitances or the inductances amalgamated with the C4 bumps, which is used to connect V_{dd} and ground connections.

4.2.3 PGN Analysis

The main motive of performing power grid analysis (or IR drop analysis) is to find the voltage affected nodes created by the IR drop. Generally power grid analysis is done using steady state analysis and transient analysis. Initially steady state analysis is done to see the steady state current

and voltages of the circuit. To see the transient changes in currents and voltages, transient analysis is also done. Here, we are considering only the steady state model for our problem formulation and experiments as from the steady state model we can know about primitive nature of the circuit. We represent the steady-state model of the PGN as a system of linear equations i.e., $\mathbf{GV} = \mathbf{I}$, where $\mathbf{G}$ matrix denotes the conductances of the metal lines, the current sources are interpreted as $\mathbf{I}$ vector and node voltages of all the nodes of the grid form the $\mathbf{V}$ vector. We adapt a direct solver as proposed in [12] named as KLU solver, for solving the matrices and obtaining the current, voltages of the PGN. We use this current and voltages as input to our multiobjective optimization engine.

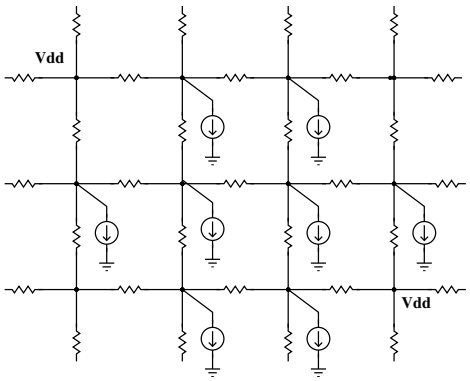


Figure 4.3: Resistive network model of PGN.

4.2.4 PGN Optimization

The objective of the PGN optimization is to minimize all the reliability issues especially to minimize IR drop which may occur in the power grids. Generally, after locating the IR drop affected nodes different approaches can be adopted to reduce the IR drop. In this work, we have adopted widening the wire length method as it doesn't require any change in the topology of the grid. Accordingly we tried to automate the IR drop minimization process along with considering metal routing area as another objective and formed an multiobjective optimization problem under the reliability constraints.

4.2.5 Background of Single and Multi-objective Optimization

In order to make sure basic understanding, few terms related to the single and multi-objective optimization method are defined for the completeness of the chapter. For single objective optimization problem, we have only one objective to optimize subject to some constraints. Single objective problems can be optimized using any search or heuristic technique. A multi-objective optimization problem has two or more contradicting objective functions. The idea is to obtain an optimum trade-off point

satisfying all the objective functions and the corresponding equality and inequality constraints. In mathematical term, it can be expressed as follows

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{minimize}} \quad & \{f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \cdots, f_k(\boldsymbol{x})\},\ k \geq 2 \\ \text{subject to} \quad & g_i(\boldsymbol{x}) \leq b_i,\ i = 1, \ldots, m, \end{aligned} \tag{4.1}$$

where we have k objective functions$f_i : \mathbb{R}^n \to \mathbb{R}$ bounded by m constraint functions g_i. $\boldsymbol{x}$ represents the decision variable vector which is generally independent of the optimization problem and belongs to the decision vector set (space) $S \subset \mathbb{R}^n$ such that $\boldsymbol{x} \in S$. The decision variable space S comprises the area that is to be searched during the optimization process. S covers all the potential values the decision variables can consider which is shown in Figure 4.4, where feasible decision variable space S' represents all the feasible values the decision variables can assume multi-objective functions with two objectives as f_1 and f_2 under m constraint functions.

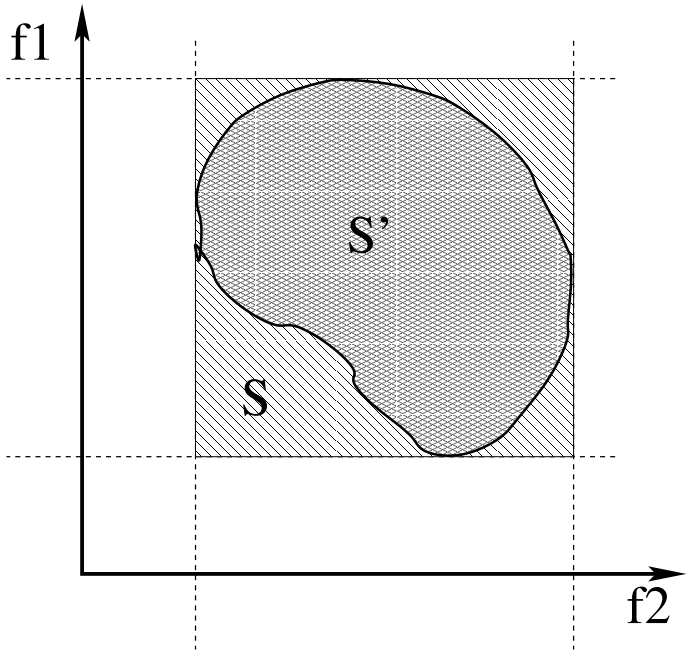


Figure 4.4: Decision variable and search space

4.2.5.1 Pareto-Optimal Solution

Definition 4.1. *[78] A decision vector $\boldsymbol{y}^* \in S'$ can be called Pareto Optimal if and only if there does not exist another decision variable vector $\boldsymbol{y} \in S'$ such that $f_i(\boldsymbol{y}) \leq f_i(\boldsymbol{y}^*)\ \forall i = 1, 2, \cdots, k$ and $f_j(\boldsymbol{y}) < f_j(\boldsymbol{y}^*)$ for at least one index of j.*

From the definition it is clear that a multiobjective problem generates Pareto-optimal solutions or non-inferior solutions and none of the solutions is stated as better than other solutions. These

Pareto-optimal solutions give rise to Pareto fronts.

4.2.6 Motivations behind using evolutionary algorithm

There are many classical methods for solving multiobjective optimization algorithms for example weighted sum methods, ϵ-Constraint method, weighted metric methods, Benson's method, value function method and many iterative methods [79]. However these methods have many disadvantages in convergence while solving non-convex functions [79]. Solutions of the non-convex functions can be achieved by minimizing a *Tchebycheff metric* (Chebyshev metric) constructed by using multiple objectives which also requires weight vector to handle objectives differently [78]. Most of the classical algorithms convert the multiobjective problem into a single objective problem by introducing some user-defined parameters [79]. For some of these algorithms, it has been proved that optimal solution of the converted single objective problem is also one of the Pareto-optimal solution [78]. However, their practical use needs many iterations of these algorithms to achieve a optimal solution. Moreover, these classical algorithms involve a number of user-defined parameters. Therefore, evolutionary algorithm has been used to solve our multiobjective optimization problem of which one of the objective is non-convex by nature.

4.3 Single Objective Problem Formulation: IR Drop Minimization

4.3.1 Objective Function for IR Drop Minimization

Let's consider $G = \{V, E\}$ be a graph corresponding to a power grid network, where $V = \{1, 2, \cdots, n\}$ is the set of all the n nodes of the power grid network and $E = \{1, 2, \cdots, b\}$ is the set of all the b branches of the graph corresponding to the steady state model of power grid network.

If I is the current passing through a metal segment (branch of the graph) of the Power Grid network having resistance R, then the voltage drop occurred across the metal segment can be represented by υ,

$$\upsilon = IR \tag{4.2}$$

With sheet resistance ρ $\Omega/\square$ which is constant for a metal layer, having metal segment length and breadth of l and w respectively, the voltage drop can be denoted by the following:

$$\upsilon = I\frac{\rho l}{w}, \tag{4.3}$$

where $R = \frac{\rho l}{w}$ represents the total resistance of the metal segment of the power line. Similarly, the

voltage drop of the whole Power Grid network with b number of metal wire segments (or branches) can be written as follows:

$$\begin{aligned} \upsilon &= \sum_{i=1}^{b} |I_i| \, R_i \\ &= \sum_{i=1}^{b} |I_i| \, \frac{\rho l_i}{w_i}, \end{aligned} \tag{4.4}$$

where $\mathbf{W}, \mathbf{I}, \mathbf{l}$ are set of vectors of metal widths, branch currents and metal lengths respectively i.e., $\mathbf{W} = (w_1, \cdots, w_b)$, $\mathbf{I} = (I_1, \cdots, I_b)$, $\mathbf{l} = (l_1, \cdots, l_b)$. For large value of b, equation (4.4) can be treated as the large scale optimization problem. In equation (4.4), I_i and w_i are the variables for the i^{th} metal segment which are non-separable in nature. Non-separable variables are those for which objective function depends on the interacting variables [80]. l_i has been taken as constant for the objective function throughout this work which can be imported from the circuit netlist. Therefore, for the whole power grid network, vectors $\mathbf{I}$ and $\mathbf{W}$ are sets of non-separable variables. Hence, the objective function can be formulated as a large-scale optimization problem with b non-separable variables as follows:

$$\upsilon(I_i, w_i) = \sum_{i=1}^{b} |I_i| \, \frac{\rho l_i}{w_i} \tag{4.5}$$

Hence the IR drop minimization problem with unequal branch currents I_i is given as follows:

$$\mathscr{P} : \underset{w_i \in W \;\; I_i \in I}{minimize} \; \upsilon, \tag{4.6}$$

subject to different reliability constraints which are described in section 4.3.2.

Theorem 4.1. *Minimization of total IR drop υ of (4.5) reduces the worst case (maximum) IR drop.*

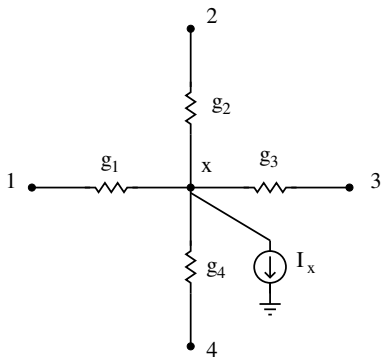


Figure 4.5: PGN model with only one node connected to a current source.

Proof. The worst case maximum IR drop $\upsilon_{\text{IR max}}$ can be expressed as

$$\upsilon_{\text{IR max}} = Max(V_{DD} - V_x)\ \forall x \in V, \tag{4.7}$$

where V_x is the node voltage of the x^{th} node which depends on the voltage of the neighboring nodes and also depends on the current of the neighboring edges. Using KCL for a node V_i (see Figure 4.7), carrying current I_i from x to i, the expression of V_x can be written as follows:

$$\begin{aligned} I_i &= \frac{V_x - V_i}{R_i}\ \forall i \in K_s \\ \Rightarrow V_x &= V_i + I_i R_i \\ \Rightarrow V_x &= V_i + I_i \frac{\rho l_i}{w_i}, \end{aligned} \tag{4.8}$$

where K_s is the set of neighboring nodes of x. Therefore, V_x of (4.8) depends on the neighboring node voltages, neighboring branch currents and resistances. Also, υ of (4.4) depends on neighboring branch currents and resistances. Therefore, minimizing υ of (4.4) under the constraint $\mathcal{C}_1 : |I_{i\in E}|\, R_{i\in E} \leq \xi \quad \forall i \in E$ by varying current and resistance reduces the worst case maximum IR drop $\upsilon_{\text{IR max}}$ of (4.7). □

4.3.2 Constraints for IR drop minimization

4.3.2.1 IR Drop Constraints

It can be defined by the following relation:

$$\mathcal{C}_1 : |I_{i\in E}|\, R_{i\in E} \leq \xi \tag{4.9}$$

The above relation should be maintained for all the i^{th} branches of the power grid network. ξ is the maximum tolerance level of voltage drop noise allowed between two consecutive nodes of the power grid network.

4.3.2.2 Metal Area Constraints

The metal area of the power grid network should be restricted to $\mathcal{A}_{max}$:

$$\mathcal{C}_2 : \sum_{i=1}^{b} l_i w_i \leq \mathcal{A}_{max} \tag{4.10}$$

4.3.2.3 Electromigration Constraints

To prevent the current carrying metal lines from electromigration, the current density of the metal lines should be less than I_m

$$\mathcal{C}_3 : \frac{I_{i\in E}}{w_{i\in E}} \leq I_m \tag{4.11}$$

4.3.2.4 Minimum Width Constraints

The minimum width of the metal lines w_{min} is limited by the technology on which the power grid network lies. Therefore, the constraint can be expressed as:

$$\mathcal{C}_4 : w_{i \in E} \geq w_{min} \tag{4.12}$$

4.3.2.5 KCL Constraints

KCL should be followed at all the n nodes of the power grid network.

$$\mathcal{C}_5 : \sum_{i=1}^{K} I_{j_i} + I_x = 0 \ \forall j \in V \tag{4.13}$$

where K is the number of neighboring nodes of node j and I_x is the sink current of the model connected to ground.

4.3.3 Minimization using Cooperative Coevolution

For the minimization, we have employed Cooperative Coevolution (CC) based evolutionary algorithm which is described in Appendix B. CC is introduced into Genetic Algorithm for optimization of function by Potter et al. [81]. Liu et al. [82] used CC in large-scale optimization problem by using Fast Evolutionary Programming with Cooperative Coevolution. CC is introduced into PSO by Bergh et al. [83]. CC has also been adapted into Differential Evolution(DE) in [84], [80]. An improved version of DE is Self-Adaptive Differential Evolution with Neighborhood Search(SaNSDE) [85] which self-adapts its scaling factor F, crossover rate CR, and mutation strategy. It is proved that SaNSDE performs quite well compared to the other similar DE algorithms [85]. Yang et al. [80] showed that SaNSDE under CC framework (CC-SaNSDE) for large-scale variable optimization works very well. To deal with the non-separable nature of the problem, random grouping based decomposition strategy of the decision variables is used. Generally, in large-scale problems, only a proportion of variables interact with each other, therefore, the random grouping of variables increase the probability of grouping two interacting variables in the same subcomponent [86]. So CC-SaNSDE has been adapted here to solve total IR drop minimization of power grid network.

4.3.4 IR drop minimization using CC-SaNSDE

The IR drop minimization algorithm using CC-SaNSDE is given in Algorithm 4.1. For the problem $\mathscr{P}$, number of variables are decomposed to form subcomponents and then each of the subcomponents is

Algorithm 4.1: IR drop minimization using CC-SaNSDE

Input: Both branch width & current ranges (for problem $\mathscr{P}$) are given as input from a power grid circuit netlist with b branches and n nodes. Branch lengths from the netlist are also taken as input.

Output: Optimum power grid netlist having minimized IR drop along-with the corresponding optimum resistance budget and metal width budget for the b branches.

1 Search space S is constructed in such a way to incorporate the reliability constraints $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_3$, $\mathcal{C}_4$, and $\mathcal{C}_5$ mentioned in section 4.3.2.;
2 **while** *inside search space* S **do**
3 Initialize the initial parameters for CC-SaNSDE with random grouping.;
4 Random grouping is employed to decompose the b variables in t subcomponents.;
5 SaNSDE optimization algorithm is used to optimize each of the subcomponents.;
6 Random grouping strategy is also used for the co-adaptation of all the subcomponents.;
7 Optimum metal widths corresponding to minimized IR drop is found and model parameters are updated.;
8 KLU solver is used to find the optimum IR drop.;

minimized using SaNSDE. The subcomponents are formed based on random grouping of variables and each of the subcomponents are minimized using SaNSDE. And finally, random grouping based strategy is used for co-adaptation of the subcomponents. Also, SaNSDE has been modified to incorporate the reliability constraints as mentioned in section 4.3.2, to keep the search space within the region of validation. Branch widths are calculated corresponding to the resistances of the branches and ranges of branch width is given as input for the problem $\mathscr{P}$. Apart from the branch width ranges, power grid analysis is done using KLU solver [12] to find the branch current ranges of the power grid network and given as input.

4.3.5 Experimental Results

Experiments are performed on standard IBM power grid benchmarks (refer to Appendix A), which demonstrate that IR drop minimization is obtained at the expense of increase in metal routing area, as listed in Table 4.1. Since we have got an increase in metal routing area of power grid network.

Table 4.1: Comparative Study of Proposed IR Drop Minimization before optimization phase

PG_circuits	Before optimization	After Optimization		
	$V_{\text{IR worst initial}}$ (V)	$V_{\text{IR worst}}$ (V)	$\Delta V_{\text{IR worst}}$	ΔA
*ibmpg*2	0.0369	0.0263	-28.72%	+17.87%
*ibmpg*3	0.2438	0.1879	-22.92%	+14.33%
*ibmpg*4	0.0086	0.0041	-52.32%	+22.65%
*ibmpg*5	0.0690	0.0431	-37.53%	+18.27%
*ibmpg*6	0.2063	0.1529	-25.88%	+15.82%

Therefore, in the next section we have performed the area minimization forming it as single objective optimization problem.

4.4 Single Objective Problem Formulation: Metal Routing Area Minimization

4.4.1 Objective Function for Metal Routing Area Minimization

Here we consider our PGN as a graph $G = \{V, E\}$ with all the nodes of the PGN as vertices set $V = \{1, 2, \cdots, n\}$ and all the branches of the PGN as edges set $E = \{1, 2, \cdots, b\}$ for the DC load model of the PGN. A pictorial representation of metal lines of 3×3 PGN is shown in Fig. 4.6

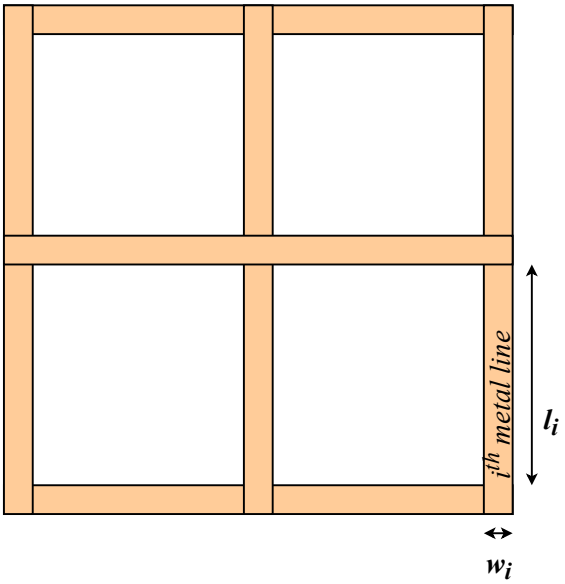


Figure 4.6: A pictorial representation of metal lines of 3×3 PGN

If l (length) and w (width) are the dimensions of a single metal fragment of the PGN which exhibits resistance R, then area covered by the metal fragment is expressed as A :

$$A = lw \tag{4.14}$$

For the same metal fragment, if it exhibits a sheet resistance $\rho\ \Omega/\square$ which is generally considered to be constant for the same layer of the metals, then the resistance of the metal fragment is analytically expressed as:

$$R = \frac{\rho l}{w}, \tag{4.15}$$

For a current of I A across the metal line, the voltage drop (IR drop) across the metal line can be

defined by,

$$\begin{aligned} V_{ir} &= IR \\ &= I\frac{\rho l}{w} \end{aligned} \tag{4.16}$$

Also the power dissipation of a metal line is represented by the following

$$\begin{aligned} P_{diss} &= IV_{ir} \\ &= I^2R \end{aligned} \tag{4.17}$$

Our objective here is the minimization of the metal routing area of the PGN maintaining IR drop (V_{ir}) within an acceptable limit, with having less power dissipation (P_{diss}), and also subject to other reliability constraints mentioned in the section 4.4.2. Hence, for the entire PGN containing b metal wire fragments (or edges) the total metal routing area is expressed as given below:

$$A_{total} = \sum_{i=1}^{b} l_i w_i \tag{4.18}$$

A large PGN has a large value of b, which makes (4.18) a cost function containing a large tally of decision making variables. In view of this, as the cost function of (4.18) has to be minimized, hence this cost function is termed as large-scale minimization problem, where w_i makes the variables set $\mathbf{w} = (w_1, w_2, \cdots, w_b)$ for $i = 1, 2, \cdots, b$. Here, l_i is considered to be a constant for the cost function (equation (4.18)) and the value of l_i is imported from the PGN netlist in order to evaluate the cost function. Therefore, the cost function is expressed as a large-scale total metal routing area minimization problem with b number of variables and is constructed as follows:

$$\mathscr{P} : \underset{w_i \in W}{minimize}\ A_{total}, \tag{4.19}$$

subject to the constraints mentioned in the Section 4.4.2.

4.4.2 Constraints for metal routing area minimization

4.4.2.1 IR Drop Constraints

From (4.16) the IR drop restriction is established by the expression given below::

$$\mathcal{C}_1 : |I_{i\in E}|\, \rho \frac{l_{i\in E}}{w_{i\in E}} \leq \xi \tag{4.20}$$

The inequality given above should be strictly obeyed for all the i^{th} edges of the PGN. ξ is the highest value of tolerance of IR drop noise permitted between two connected vertices of the PGN. Basically ξ is the maximum allowable voltage difference between two consecutive nodes of the PGN.

4.4.2.2 Metal Line Area Constraint

In order to limit our design in a confined area, the total metal routing area occupied by the metal lines of the PGN should be limited to $\mathcal{A}_{max}$:

$$\mathcal{C}_2 : \sum_{i=1}^{b} l_i w_i \leq \mathcal{A}_{max} \tag{4.21}$$

4.4.2.3 Current Density Constraint

The maximum current density of the metal lines of the PGN should be limited to I_m, in order to avoid degradation of the metal lines due to electromigration based reliability issues.

$$\mathcal{C}_3 : \frac{I_{i\in E}}{w_{i\in E}} \leq I_m \tag{4.22}$$

4.4.2.4 Metal Line Width Constraint

The design of the metal lines should follow the design rules of the given CMOS technology nodes and should follow the minimum width design rules in order to avoid any design rule violations. The metal width constraint can be represented as follows:

$$\mathcal{C}_4 : w_{i\in E} \geq w_{min} \tag{4.23}$$

4.4.2.5 Current Conservation Constraint

At all the n vertices of the PGN, Kirchhoff's Current Law (KCL) or the current conservation constraint must be observed, which is represented as follows:

$$\mathcal{C}_5 : \sum_{i=1}^{K} I_{j_i} + I_x = 0 \ \forall j \in V \tag{4.24}$$

where the symbol K denotes neighboring vertices tally around the vertex j and the symbol I_x represents DC load current of the PGN model which is placed at all node of nodes to the ground.

4.4.2.6 Power Dissipation Constraint

The power dissipation of a metal interconnect of the PGN should be limited by ψ.

$$\begin{aligned} \mathcal{C}_7 : P_{diss} &= I_i V_{i_{ir}} \leq \psi \\ &= I_i^2 R_i \leq \psi \\ &= \frac{\rho l_i I_i^2}{w_i} \leq \psi \end{aligned} \tag{4.25}$$

Property 4.1. *Minimization of area of the PGN is dependent on reliability, and power-aware constraints*

Proof. From equation (4.18), we know that A_{total} depends upon width (w_i) of each of the metal interconnect.

$$A_{total} \propto w_i \tag{4.26}$$

Therefore, to minimize the area we have to reduce the w_i. However, most of the constraints mentioned in section 4.4.2 directly or inversely proportional to w_i

$$Constraints \ \propto w_i \ or \frac{1}{w_i} \tag{4.27}$$

and reducing w_i surely affects these constraints. Therefore, minimization of the area depends on the reliability, and power-aware constraints. □

4.4.3 Metal Area minimization using CC-SaNSDE

Algorithm 4.2: Metal Area minimization using CC-SaNSDE

Input: The cost function $\mathscr{P}$, widths (w_i), lengths (l_i), number of edges (b), number of vertices (n). All the data is extracted from the power grid netlist.
Output: Optimum metal widths of the metal lines with decreased total metal routing area.

1 The reliability, and power-aware constraints $\mathcal{C}_1$, $\mathcal{C}_2$, $\cdots$, $\mathcal{C}_7$ mentioned in section 4.4.2 are incorporated to generate a search space T.;
2 **while** *inside search space* T **do**
3 Initialization is done for the initial parameters of CC-SaNSDE;
4 Decomposition of the b variables in t subcomponents is done using random grouping strategy;
5 For optimizing the subcomponents, SaNSDE optimization algorithm is used;
6 The subcomponents are co-adapted by randomly grouping the best solutions of the subcomponents.;
7 Optimum widths of the metal lines of the PGN is evaluated corresponding to the minimized total metal routing area and the model parameters are updated.;

For this minimization problem also we employ CC-SaNSDE, because of the similar nature of the problem. The total metal routing area minimization algorithm for PGN exercising CC-SaNSDE is presented in Algorithm B.2. Initially, to find all the edge currents and all node voltages of the PGN, the grid analysis of PGN is performed with the help of the KLU based matrix solver [12]. In order to power grid analysis, KLU Solver is used subsequently to solve the linearized system of equations of the

PGN. All the required parameters are initialized for the SaNSDE. Search space T is created considering all the constraints mentioned in section 4.4.2 to restrict the search space of the cost function interior to the area of validation. Subsequently for the problem $\mathscr{P}$, initially Cooperative Coevolution based scheme is used to decompose a large number of variables of the problems into smaller instances of the subcomponents. The decomposed variables are collected randomly in a number of smaller groups to make different subcomponents. Once the subcomponents are formed, the minimization of each of the subcomponents is done individually, with the help of SaNSDE optimizer. Eventually, after all the subcomponents are minimized individually, again arbitrary grouping based co-adaptation scheme is employed to reach the global near-optimum value of the cost function. In this way, the total metal routing area of $\mathscr{P}$ is minimized. Optimized edge widths are determined corresponding to the minimized metal routing area of the PGN with the help of the cost function $\mathscr{P}$.

4.4.4 Experimental Results

Experiments are performed on standard IBM power grid benchmarks, which demonstrate that metal routing area minimization is obtained at the expense of increase in worst-case IR drop, as listed in Table 4.2. Further, in order to obtain a trade-off between IR drop and metal routing area of on-chip

Table 4.2: Comparative Study of Proposed Metal Routing Area Minimization before optimization phase

PG_circuits	Before optimization	After Optimization		
	Area ($10^6 \mu m^2$)	Area ($10^6 \mu m^2$)	ΔA	$\Delta V_{\text{IR worst}}$
*ibmpg*2	1449.19	1130.36	-22.02%	+14.17%
*ibmpg*3	3462.86	2576.35	-25.61%	+12.56%
*ibmpg*4	6162.69	4940.01	-19.84%	+16.19%
*ibmpg*5	2870.70	2263.54	-21.15%	+15.33%
*ibmpg*6	4505.54	3393.12	-24.69%	+13.21%

power grid network, the following multi-objective formulation is constructed.

4.5 Multiobjective Problem Formulation: IR Drop and Area as Conflicting Objectives

4.5.1 Problem Formulation

We assume the PGN as a graph $G = \{P, Q\}$, where $P = \{1, 2, \cdots, n\}$ represents the set of nodes and $Q = \{1, 2, \cdots, b\}$ represents the set of branches equivalent to the PGN steady-state model, shown in Fig. 4.3.

Table 4.3: Parameters along with their meanings

Metric	Meaning	Value
l_i	Length of metal line i	Depends on design
w_i	Width of metal line i	
g_i	Conductance of metal line i	
R_i	Resistance of metal line i	
I_i	Current through metal line i	
I_x	Current sink of the power grid model connected to ground	
ρ	Sheet resistance of metal line	0.02 $\Omega/\square$
A_i	Area of metal line i	
V_x	Voltage of node x	
v_{*_x}	IR drop of node x	
v_{IR}	IR drop across metal line i	
$\%\vartheta$	Percentage of affected nodes above V_{th}	
$V_{tolerance}$	Tolerance level of voltage	$10\% of V_{dd}$
V_{th}	Threshold voltage	
V_{dd}	Supply voltage	1.8V
ξ	Voltage difference between two consecutive node	

If I is the steady-state current crossing within a metal segment of the PGN having resistance R, then the IR drop or the voltage drop (v_{IR}) generated across the metal segment can be calculated as follows :

$$v_{\text{IR}} = IR, \tag{4.28}$$

where R denotes resistance of the metal segment and I is steady-state current through the segment. The voltage drop can be written in terms of sheet resistance ($\rho\ \Omega/\square$), metal wire length (l) and width (w) as follows,

$$v_{\text{IR}} = I\frac{\rho l}{w} \tag{4.29}$$

where $R = \frac{\rho l}{w}$ denotes the total resistance of the metal segment of the power line. Furthermore, the total IR drop or voltage drop of the complete PGN having b metal line segments (or branches) can be formulated as given below:

$$\begin{aligned} v_{\text{IR total}} &= \sum_{i=1}^{b} |I_i|\, R_i \\ &= \sum_{i=1}^{b} |I_i|\, \frac{\rho l_i}{w_i} \end{aligned} \tag{4.30}$$

where the notations $\mathbf{W}, \mathbf{I}, \mathbf{l}$ are vectors which represents metal segment widths, metal segment currents and metal segment lengths respectively i.e., $\mathbf{W} = (w_1, \cdots, w_b)$, $\mathbf{I} = (I_1, \cdots, I_b)$, $\mathbf{l} = (l_1, \cdots, l_b)$.

To minimize the IR drop, width of the metal segments need to be increased. However by doing so, we are increasing the metal wires area, which decreases yield of the chip, so we need to minimize the area also. Therefore objective function for the area can be written as follows:

$$A = \sum_{i=1}^{b} l_i w_i \tag{4.31}$$

where $A_i = l_i w_i$ is area of the i^{th} metal wire segments, with l_i as length and w_i width of the metal segments.

The aforementioned problem can be formulated as the multi-objective optimization problem as follows:

$$\begin{aligned}
\underset{I_i, w_i}{\text{minimize}} \quad & \begin{cases} \upsilon_{\text{IR total}} = \sum_{i=1}^{b} |I_i| \, \frac{\rho l_i}{w_i} \\ A = \sum_{i=1}^{b} l_i w_i \end{cases} \\
\text{subject to} \quad & |I_i| \, \frac{\rho l_i}{w_i} \leq \xi \ \forall i \in Q, \qquad\qquad l_i w_i \leq A_{seg} \ \forall i \in Q, \\
& \sum_{i=1}^{b} A_i \leq \mathcal{A}_{max}, \qquad\qquad w_i \geq w_{min} \ \forall i \in Q, \\
& \sum_{i=1}^{K} I_{j_i} + I_x = 0 \ \forall j \in P \ \ \text{(KCL)}, \\
& I_i \leq I_{max}.
\end{aligned} \tag{4.32}$$

where, KCL stands for Kirchoff's Current Law, which describes the conservation of electric charge in every node of an electric circuit. A_{seg} is the maximum area allowed for each metal segment. A_{max} maximum total metal routing area allowed for the design. I_{max} is the maximum current density of each of the interconnect. i represents the indices of the interconnects and j represents the indices of nodes. The formulated multi-objective optimization problem is to minimize the two contradicting objectives of PGN design phase IR drop and area simultaneously.

Corollary 4.1. *The total IR drop $\upsilon_{IR\ total}$ under the constraint $\mathcal{C}_1 : |I_i| \, \frac{\rho l_i}{w_i} \leq \xi \ \forall i \in Q$ (mentioned in section 4.5.2.1) is directly depends on the worst case maximum IR drop $\upsilon_{IR\ max}$. The reduction of total IR drop $\upsilon_{IR\ total}$ also reduces the worst case maximum IR drop.*

From the Corollary 4.1, it is clear that minimization of total IR drop also reduces the worst case IR drop. To be more specific, minimizing total IR drop reduces the effects of IR drop at all the nodes. The number of nodes affected by the IR drop above a threshold voltage V_{th} represented by ϑ, also reduces

due to minimization of total IR drop. Percentage reduction of nodes can be expressed as follows:

$$\%\vartheta = \frac{|P| - |v_*|}{|P|} \times 100, \tag{4.33}$$

where v_* is the set of affected nodes by the IR drop above V_{th}. Let, $v_{\text{IR node}}$ be a element of v_* which can be calculated as below:

$$v_{\text{IR node}} = V_{DD} - V_x; \quad V_x \geq V_{th} \tag{4.34}$$

Therefore, calculating ϑ requires the node voltages of all the nodes. For a node x, an alternative expression of V_x using KCL can be given as below:

$$\begin{aligned} I_x &= \sum_{i=1}^{K} (V_x - V_i) g_i \\ \Rightarrow V_x &= \frac{\sum_{i=1}^{K} g_i V_i}{\sum_{i=1}^{K} g_i} + \frac{I_x}{\sum_{i=1}^{K} g_i}, \end{aligned} \tag{4.35}$$

where $g_i = \frac{1}{R_i}$ and I_x is the current sink from the node x of the power grid model going to the ground as shown in Figure 4.7 and K is the adjacent nodes count of the node x. V_i also depends on the adjacent branch currents I_i. Since $\%\vartheta$ and A both depends on I_i and R_i and both objectives

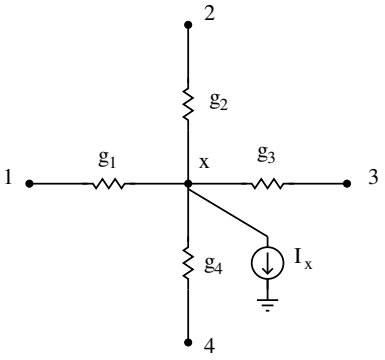


Figure 4.7: PGN model with only one node connected to a current source.

are conflicting. Therefore, under the constraints mentioned in (4.32), the multiobjective minimization problem of (4.32) can also be rewritten as:

$$\underset{I_i, w_i}{\text{minimize}} \begin{cases} \%\vartheta \\ A \end{cases} \tag{4.36}$$

4.5.2 Multi-objective Reliability and Yield Constraints of PGN

The constraints mentioned in (4.32) can affect the objective functions so here the constraints are described concisely:

4.5.2.1 IR drop Reliability Constraint:

The voltage drop constraints can be described by the following relation:

$$\mathcal{C}_1 : |I_i| \frac{\rho l_i}{w_i} \leq \xi \ \forall i \in Q \tag{4.37}$$

The above expression has to validated for all the i^{th} interconnects of the power network of the PGN. ξ is the voltage drop difference between two consecutive nodes.

4.5.2.2 Metal Area-based Yield constraints:

The total metal routing area should be restricted to $\mathcal{A}_{max}$ and metal routing area of each of the branches should be restricted to A_i

$$\mathcal{C}_2 : l_i w_i \leq A_{seg} \ \forall i \in Q \tag{4.38}$$

$$\sum_{i=1}^{b} A_i \leq \mathcal{A}_{max} \tag{4.39}$$

4.5.2.3 Electromigration Reliability constraint:

To prevent the power grid lines from electromigration-based reliability failure, the current density of the grid lines should be limited by I_{max}

$$\mathcal{C}_3 : I_i \leq I_{max} \ \forall i \in Q \tag{4.40}$$

4.5.2.4 Minimum metal line width constraints:

The minimum width of power grid lines w_{min} is selected depending upon the technology nodes. The constraint can be written as follows,

$$\mathcal{C}_4 : w_i \geq w_{min} \ \forall i \in Q \tag{4.41}$$

4.5.2.5 Node constraints:

KCL has to be validated at all the n nodes of the PGN in order to ensure the electrical rules check.

$$\mathcal{C}_5 : \sum_{i=1}^{K} I_{j_i} + I_x = 0 \ \forall j \in P \tag{4.42}$$

where K is the number of adjacent nodes of node j.

4.5.3 Constraints Handling

To handle different reliability constraints different constraint handling techniques have been used which has been mainly categorized in boundary constraint, equality constraint, and non-equality constraint.

4.5.3.1 Boundary Constraints

During evaluation of the boundary constraints, it is necessary to guarantee that the design parameter values are within the domain of all the constraints. When the design parameters go out of the domain of the constraints, then it is substituted by a random value within the domain [87]. The random value is calculated as given below [87]

$$x_c = \begin{cases} low_c + rand(0,1) \times (upp_c - low_c), \text{ if } \ x_c < low_c \ or \ x_c > upp_c \\ x_c, \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{otherwise} \end{cases} \tag{4.43}$$

where x_c denotes the random value, $rand(0,1)$ returns a real value between 0 and 1 with uniform distribution, low_c and upp_c denote the lower and upper bounds of design parameter c respectively.

4.5.3.2 Equality Constraints

We use a most probable point-based equality constraint handling method as mentioned in [88]. In this method, the optimization problem is solved to find the point on the constraint boundary. The point on the constraint boundary closest to a given solution is considered as the most probable point.

4.5.3.3 Inequality Constraints

The inequality constraints during on-chip power grid design are satisfied using the constrained-domination principle [77]. It states that feasible solutions are identified and ranked according to their non-domination level.

4.6 Proposed Minimization Framework for PGN

The minimization framework deals with the objective function mentioned in (4.32). With increase in number of nodes in the PGN model the number of decision variables of the objective function mentioned in (4.32) also increases. This can be seen as large scale multi-objective problem. Solving such multi-objective problems for millions of decision variables with millions of constraints is not computationally feasible till now as can be seen in [89] where the authors have done multi-objective optimization for only 5000 decision variables. Therefore, to solve our multi-objective problem, we have to follow divide and conquer approach. We need to minimize the IR drop-metal routing area of the PGN subcircuit by subcircuit. A subcircuit consist of a single node and four edges of the PGN is shown in Figure 4.7. So initially we'll be solving the minimization problem for a subcircuit with a single node, the neighboring edges connected to it and the current sink connected to the ground. So for a subcircuit with a single node the objective function of (4.32) can be rewritten as

$$\underset{I_i, w_i}{\text{minimize}} \quad \begin{cases} \upsilon_{\text{IR subckt}} = \sum_{i=1}^{K} |I_i| \, \frac{\rho l_i}{w_i} \\ A_{\text{subckt}} = \sum_{i=1}^{K} l_i w_i \end{cases} \tag{4.44}$$

subject to constraints mentioned in section 4.5.2, where K denotes the number of neighboring nodes of the j^{th} node which is generally 4 or 5 for PGN. NSGA-II based evolutionary algorithm [77] is adopted to solve the multi-objective IR drop-metal routing area minimization problem, as it is one of the well-known multi-objective optimization technique. Many other multi-objective optimization algorithms are also developed, and any of them can be used. Moreover, we are adapting the constraint handling techniques in NSGA-II for solving our problem. The NSGA-II algorithm is explained in the Appendix B.

4.6.1 Minimization Framework

Our proposed minimization framework is illustrated in Figure 4.8. It contains few steps for minimization of IR drop and area.

4.6.1.1 IR drop analysis using KLU Solver

Initially, power grid netlist is taken as an input to the framework. IR drop analysis is basically finding current and voltages of the circuit. Therefore, to get the all branch currents and voltages modified nodal analysis(MNA) matrices are created from the power grid netlist and KLU solver [12]

is used as it has better speedup than *Synopsys HSPICE circuit simulator* [90]. The reason The reason behind the better speedup is KLU efficiently solves the linear system of equations resulting from the modified network analysis. In KLU solver [12], the matrix is permuted in block triangular form and each block is ordered to reduce the fill. For LU factorization *Gilbert-Peierls algorithm* is used and the system is solved using block back substitution [12]. Accordingly, the circuit solved is using KLU Solver and all circuit parameters are found. From voltage and current of the circuit the affected regions by IR drop can be located.

4.6.1.2 Setup for Minimization

As the circuit is minimized subcircuit by subcircuit, therefore, input for a subcircuit is given to NSGA-II. For a subcircuit the decision variables are branch currents and branch widths, therefore, these are given as input to the NSGA-II. Branch current ranges are found from the KLU solver is given as input to the NSGA-II. Branch width ranges of all the branches are given as input to the NSGA-II, which is found from the branch resistance of the circuits with a considering an appropriate length and sheet resistance. All other initial parameters for NSGA-II have been fixed along with the number of variables and given as input to the NSGA-II.

4.6.1.3 Minimization using NSGA-II

The minimization process begins with the subcircuit for which the IR drop is maximum. The subcircuit's violated node is considered for minimization, and interconnects connected to it go through an optimization process for IR drop and area of the metal lines. NSGA-II is adapted in the optimization of the subcircuits. Data for that violated node is given as input to the NSGA-II algorithm along with all other setups for the NSGA-II algorithm. Once minimization of the violated node is performed, rest of the nodes (subcircuits) also go through the same procedure with the help of NSGA-II until all the nodes are minimized. For each of the subcircuits, the parameters are updated after optimization using the Pareto optimal solutions, which further helps in minimization of the neighboring nodes. In this way, we can achieve a good trade-off between the IR drop and metal routing area of the PGN. The minimization procedure is mentioned in the Algorithm 4.3. As all the points of the Pareto front are equally optimal, which point is to select for each of the subcircuit is described in the subsequent subsection. Only highly violated nodes can be minimized to speed up the process, leaving the mildly affected nodes. However, in order to get an optimum trade-off between IR drop and metal routing

area, it is necessary to run the algorithm for all the subcircuits. The subcircuits with violated nodes contribute towards IR drop minimization and the subcircuits without violated nodes contribute towards area minimization. In this way, when all the subcircuits go through the optimization process, we obtain the IR drop and metal routing area trade-off.

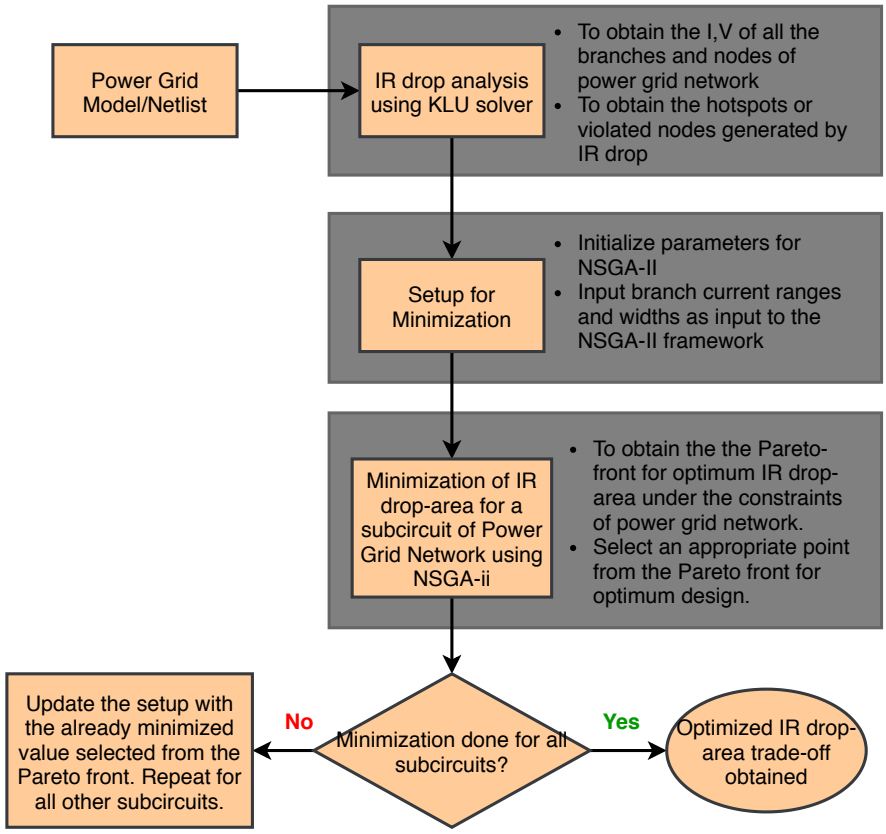


Figure 4.8: Flow of the minimization framework

4.6.1.4 Selecting Appropriate Point from Optimal Pareto Front

All points of the Pareto fronts are equally optimal. Therefore, selecting a desired optimal point from the Pareto depends upon the power grid designers. Considering the IR drop-metal routing area trade-off, an appropriate point is selected, and all parameters of subcircuit of the PGN corresponding to the selected point of the Pareto fronts are updated. In general, the knee point is selected as the appropriate point, where small improvement in one objective leads to extensive degradation in other objectives. However, for our problem, we employ criteria based on the IR drop value of the affected node of the subcircuit. If the reference node of the subcircuit is severely affected by IR drop then that point from the Pareto front is selected, which represents a minimized IR drop point (with 10-15%

Algorithm 4.3: IR drop-metal routing area minimization using NSGA-II

Input : Metal segment current ranges & width are given as input for the steady-state model of the PGN. Metal segment lengths from the netlist are also read for calculation.
Output: Total optimized IR drop along-with the corresponding optimum resistance budget, metal width budget of the b branches, and metal routing area.

Power grid netlist is divided in small subcircuits and numbering is done;
$i \leftarrow 0$;
$v_{\text{IR global}} \leftarrow v_{\text{IR subckt}_i}$;
$A_{\text{global}} \leftarrow A_{\text{subckt}_i}$;
for $subckt_{i+1}$ **do**
    Search space is constructed using the constraint handling methods described in section 4.5.3 in such a way to accommodate the reliability constraints $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_3$, $\mathcal{C}_4$, and $\mathcal{C}_5$ mentioned in section 4.5.2.;
    Initialize the initial parameters for NSGA-II algorithm;
    NSGA-II is applied to minimize the IR drop-metal routing area simultaneously for a subcircuit of the PGN model. ;
    Optimal Pareto fronts are generated for each of the subcircuits. ;
    A suitable point $(A_{subckt_{i+1}}, v_{\text{IR subckt}_{i+1}})$ from the optimal Pareto front is selected as accepted point;
    // In order to ensure, the solution doesn't stuck in local minimum
    if $v_{IR\ subckt_{i+1}} \leq v_{IR\ global}$ *and* $A_{subckt_{i+1}} \leq A_{global}$ **then**
        $v_{\text{IR global}} \leftarrow v_{\text{IR subckt}_{i+1}}$;
        $A_{\text{global}} \leftarrow A_{\text{subckt}_{i+1}}$;
        parameter values corresponding to point $(A_{\text{global}}, v_{\text{IR global}})$ is used to update the subcircuit value.;
    // Go to next subcircuit
    $i \leftarrow i + 1$

decrease in IR drop value) and metal routing area might be more in this case. If the reference node is not severely affected by IR drop then area minimized point can be selected. In the Figure 4.9, a subcircuit is shown with the reference node numbering as $n1\_5021\_14072$. For this subcircuit, its reference node is severely affected (node voltage of $n1\_5021\_14072$ is 1.45582 V, see Table 4.5), so the point (3.6e+05 μm^2, 0.0878 V) is selected (see Figure 4.10) as this point represents a minimized IR drop point (see Tables 4.5 and 4.6). After selecting the point, all the parameters, such as metal widths, branch currents, and node voltages of the subcircuit are updated. This process of minimizing the subcircuit, selecting an optimal point from the Pareto front and updating the grid parameters continue for all the subcircuits.

4.6.2 Computational Complexity

Computational Complexity of the multi-objective minimization framework is $\mathcal{O}(MN^2)$ which is also the complexity of the NSGA-II. For performing it for all the n^{th} nodes of the PGN, the complexity becomes $\mathcal{O}(nMN^2)$

4.7 Experimental Validation

4.7.1 Configuration

All the algorithms proposed are implemented in C++ language and experiments are performed on a machine with Intel Xeon E5-2650 processor having $32GB$ RAM. To demonstrate applicability and performance assessment of the minimization framework, different IBM power grid benchmarks (i.e., *ibmpg*1, *ibmpg*2, *ibmpg*4 etc) [3] are employed. The details of all benchmarks are listed in Appendix A. IBM benchmarks are generally considered for all experiments regarding power grid network in the literature as these are based on the real designs of IBM chips and easily available in Internet. Since we are only considering the case of steady-state analysis so only the DC current load benchmarks are used. All the benchmark circuits contain DC current sinks and the metal lines are modeled as resistance. Experiments are performed for the *ibmpg*1 to *ibmpg_nw*1 benchmark circuits. The metal line width and length data are not available in the IBM power grid benchmarks [3]. Therefore, relevant values of lengths are used with a the maximum length of $700\mu m$ and the equivalent width of metal layers are obtained by assuming sheet resistance of the metal $\rho = 0.02\Omega/\square$ (for 180 nm technology M6 layer) and from branch resistances using equation $R = \frac{\rho l}{w}$. IBM power grid benchmark statistics can be seen the Table A.1 (Appendix).

Setup for minimization: Inputs for NSGA-II are initialized in this step of the framework. `populationSize` is set as 100, `maxGeneration` is set as 100, `P_cross` is set as 0.9, `P_mutation` is set as 0.2. Branch width and current ranges are calculated as discussed earlier and given as input. All the reliability constraints mentioned in Section 4.5.2 are incorporated and the search space is constructed according to that. Subsequently, random population of 100 size is created to find the optimum solution of IR drop and area using NSGA-II. More details about the parameters configuration of NSGA-II for a subcircuit can be found in Table 4.4.

Minimization using NSGA-II: The minimization process starts with minimization of subcircuit. Both the objective functions of (4.44) are evaluated at the feasible points generated in the previous step for the subcircuit. These evaluated values passes through all the steps mentioned in Algorithm B.2 so as to find the optimal solutions as a Pareto front. Pareto front for a subcircuit of *ibmpg*1 (of Figure 4.9) is shown in the Figure 4.10. All solutions of the Pareto front are equally optimal, now according to the IR drop-metal routing area trade-off an appropriate point can be selected from the Pareto front. From the Pareto front of the subcircuit shown in Figure 4.10, (3.6e+05,0.0878) point

Table 4.4: Parameters configuration of NSGA-II for a subcircuit

Items	Values
Population size	100
Number of generation	100
Number of objective function	2
Number of constraints	5
Number of real variables	4
Probability of crossover	0.9
Probability of mutation	0.2
Distribution index of crossover	10
Distribution index of mutation	10
Seed for random number	0.5

has been selected and corresponding optimum metal width values are updated for the subcircuit and the minimization process goes on for the next subcircuit. The process continues for all n subcircuits. Subsequently, we have a PGN with less number of worst case IR drop and occupy less metal routing area. Comparison of worst case IR drop before and after minimization is shown in Table 4.7. Total metal routing data before and after optimization is shown in Table 4.8.

4.7.2 Results

4.7.2.1 Results for a subcircuit of PGN

This section lists results related to subcircuit of Figure 4.9. It has been demonstrated to get an idea how the subcircuit parameters are changing with the optimization process. The Pareto front for the subcircuit of Figure 4.9 has been shown in Figure 4.10. All points of the Pareto-front are equally optimal. From the Table 4.5 we can see that potential of the node $n1\_5021\_14072$ is 1.45582 which means this is a violated node. For such a violated node, the corresponding subcircuit should be minimized for IR drop. Therefore, such a point from the Pareto front should be selected, which has 10-15% less total IR drop for the subcircuit. So, we selected the point (3.6e+05, 0.0878) from the Pareto front, which corresponds to a decrease in 15% total IR drop for the subcircuit in concern. This point also corresponds to the knee point of the Pareto front. Once we select this appropriate point and update the subcircuit parameters, we can observe that the IR drop of the updated subcircuit is reduced. This is achieved at the expense of increase in metal routing area of this subcircuit. However, there are some subcircuits having no violated nodes, for such subcircuits we obtain an metal routing area reduction. Overall when we run the algorithm for all subcircuits, we achieve an optimum trade-off between IR drop and metal routing area. It is to be noted that the node voltages of the subcircuit demonstrated

in Tables 4.5 and 4.6 are quite low even after optimization. That is because this particular *ibmpg*1 benchmark circuit suffers from a high max IR drop (worst-case IR drop) of 0.9995 (see Table 4.7). Such instances do not occur for rest of the IBM benchmark circuits. Results for subcircuits are included for better understanding of the internal updates of subcircuit parameters.

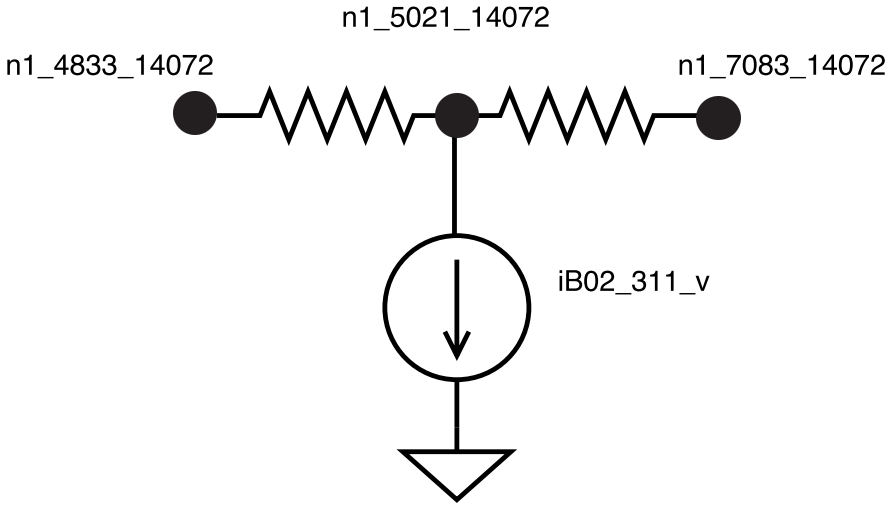


Figure 4.9: subcircuit of *ibmpg*1 circuit

Table 4.5: Data of subcircuit shown in the Figure 4.9 of *ibmpg*1 before optimization.

Parameters	neighboring branches data of subcircuit	
Neighboring node of $n1\_5021\_14072$	$n1\_4833\_14072$	$n1\_7083\_14072$
Resistance (Ω)	1.074286e+00	1.178286e+01
Widths (μm)	20	20
Length (μm)	1.0743e+03	1.1783e+04
Branch currents (A)	0.0127	-0.0076
Neighboring node voltages (V)	1.46951e+00	1.36614e+00
Node voltage of $n1\_5021\_14072$ (V)	1.45582e+00	
IR drop of branches (V)	0.01368	0.08968
	subcircuit data before optimization	
Total Metal Area of subcircuit (A_{subckt})	2.5714e+05 μm^2	
Total IR drop of subcircuit ($v_{\text{IR subckt}}$)	0.1034 V	

Table 4.6: Data of subcircuit shown in the Figure 4.9 of *ibmpg*1 after optimization for the selected point (3.6e+05, 0.0878) from Pareto front.

Parameters	neighboring branches data of subcircuit	
Neighboring node of $n1\_5021\_14072$	$n1\_4833\_14072$	$n1\_7083\_14072$
Resistance (Ω)	0.8952e+00	0.8297e+01
Widths (μm)	24	28.4
Length (μm)	1.0743e+03	1.1783e+04
Branch currents (A)	6.011243e-03	-1.000015e-02
Neighboring node voltages (V)	1.4612e+00	1.3728e+00
Node voltage of $n1\_5021\_14072$ (V)	1.45582e+00	
IR drop of branches (V)	0.00538	0.08302
	subcircuit data after optimization	
Total Metal Area of subcircuit (A_{subckt})	3.6e+05 μm^2	
Total IR drop of subcircuit ($v_{\text{IR subckt}}$)	0.0878 V	

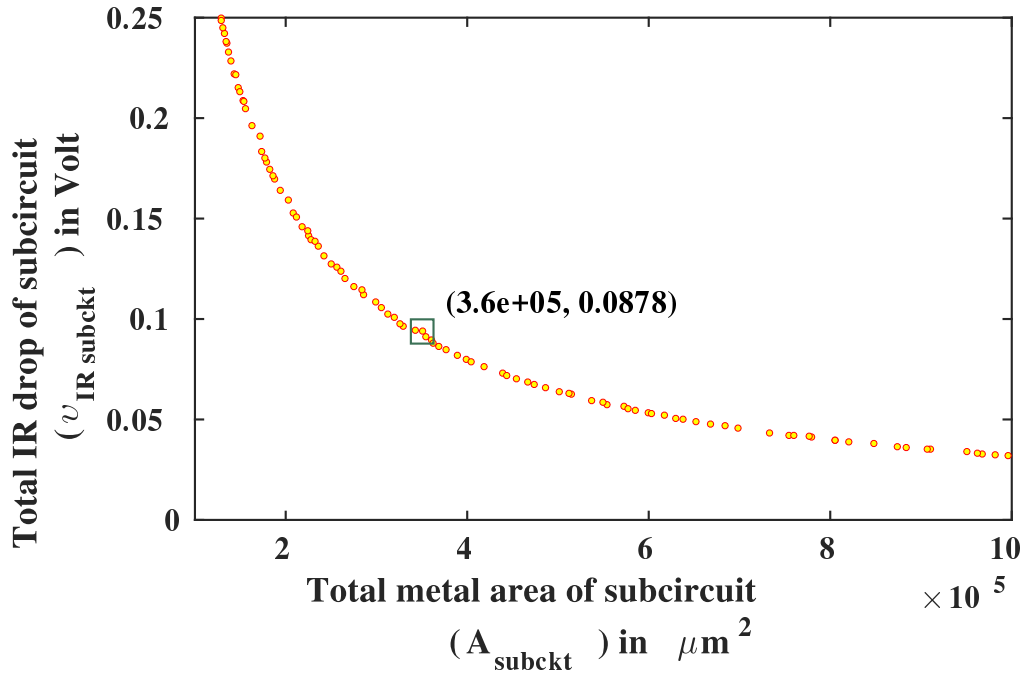


Figure 4.10: Pareto front for the subcircuit showing optimal IR drop-metal routing area trade-off

4.7.2.2 Comparison of the results before and after optimization

The optimized values for different power grid benchmarks compared to its initial values are demonstrated in this section. The same experiments are run for 10 times (with the same configuration of the evolutionary algorithms) and mean results are reported for all the parameters in this section. The result of IR drop comparison is listed in Table 4.7. From the table, we can observe that our proposed framework helps in reduction of IR drop. Our proposed framework also helps in the reduction of the metal routing area. The comparison of metal routing area before and after the optimization process is listed in Table 4.8. These two results show that the proposed NSGA-II-based framework helps in obtaining a trade-off between the critical design objectives, i.e., IR drop and metal routing area. These two optimum design objectives can help power grid designers get the initial approximated design parameters. Using which the optimum designs can be realized with very little time instead of searching for the optimum parameters. We can also observe from the Table 4.7 that as the number of nodes of the circuits increases, the execution time of the optimization process also increases. The optimization process still takes a considerable amount of time (13097 secs/3.63 hours for ibmpg6). However, this gives an initial idea to the designers about selecting different critical design parameters, which, of course, make the overall design process faster.

Table 4.7: IR drop before and after optimization and optimization time

PG_circuits	IR drop before optimization			IR drop after optimization			Time(sec)
	Max.	Avg.	Min.	Max.	Avg.	Min.	
*ibmpg*1	0.9995	0.3321	0.1099	0.8810	0.2988	0.0999	185.3
*ibmpg*2	0.0369	0.0296	0.0100	0.0342	0.0254	0.0100	652.3
*ibmpg*3	0.2438	0.1154	0.0290	0.2230	0.1038	0.0281	3458.1
*ibmpg*4	0.0086	0.0042	0.0010	0.0075	0.0037	0.0010	4763.9
*ibmpg*5	0.0690	0.0373	0.0146	0.0610	0.0335	0.0143	9301.7
*ibmpg*6	0.2063	0.0863	0.0285	0.1881	0.0776	0.0284	13097.5
*ibmpg_nw*1	0.2083	0.0902	0.0275	0.1889	0.0811	0.0274	6156.3

4.7.2.3 IR drop profile before and after optimization

The IR drop profile for both cases is shown in Figure 4.11. The IR drop profile denotes IR drop distribution across surface of the chip. In the figure, both the axes denote the axes of the chip surface. The IR drop values for power grid network is scaled for a 100×100 grid structure and plotted using python matplotlib library. Red areas in the IR drop map denote that those portions suffer from large IR drop and blue areas in the IR drop map denote that those portions suffer from least IR drop. From

Table 4.8: Total Metal routing area before and after optimization

PG_circuits	Total metal routing area ($10^6 \mu m^2$)		Area reduction (%)
	Before Optimization	After Optimization	
*ibmpg*1	234.89	213.65	9.04%
*ibmpg*2	1449.19	1326.00	8.51%
*ibmpg*3	3462.86	3178.00	8.20%
*ibmpg*4	6162.69	5674.60	7.92%
*ibmpg*5	2870.70	2652.81	7.59%
*ibmpg*6	4505.54	4175.73	7.32%
*ibmpg_nw*1	4583.89	4265.30	6.95%

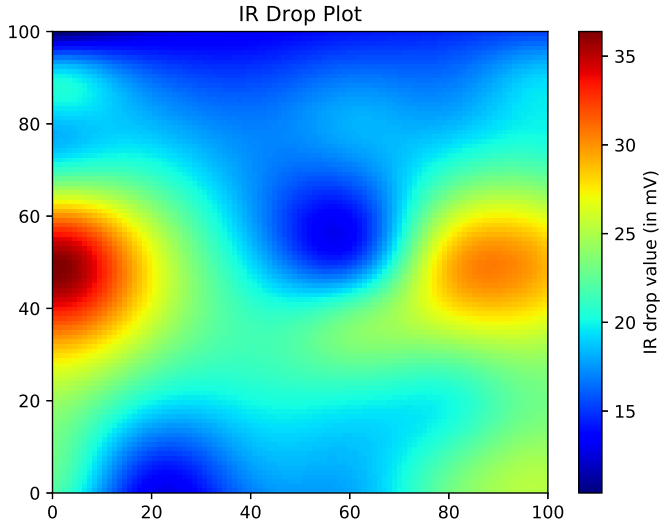
the IR drop profile, we can observe that the worst-case IR drop, as well as the average IR drop, have decreased after using the proposed framework.

4.7.2.4 Comparison of Proposed Framework with the Simple Linear Programming Approach

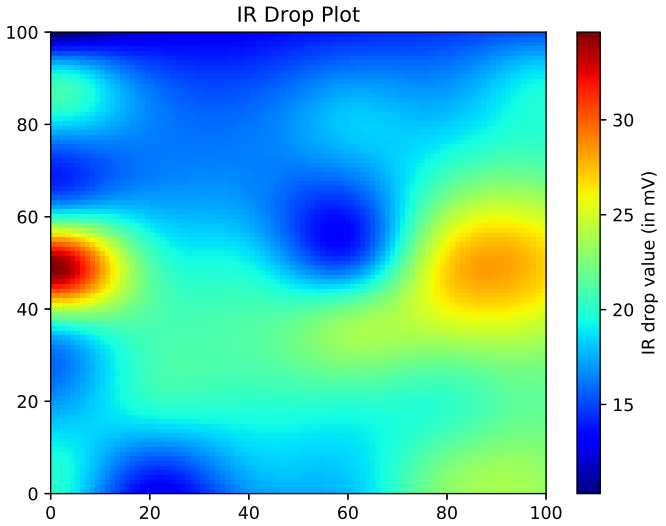
In this section, we have discussed the effectiveness of our proposed framework (using evolutionary computing) against the Simple Linear Programming (SLP) approach. For that, we need to formulate the problem as the weighted sum of the two objectives in order to demonstrate the multiobjective nature of our problem. We can rewrite (4.32) as the weighted sum form of two objectives as the following,

$$
\begin{aligned}
\underset{I_i, w_i}{\text{minimize}} \quad & (1-\eta)\sum_{i=1}^{b}|I_i|\,\frac{\rho l_i}{w_i} + \eta\sum_{i=1}^{b} l_i w_i, \quad \forall\ \eta \in [0,1] \\
\text{subject to} \quad & |I_i|\,\frac{\rho l_i}{w_i} \le \xi\ \forall i \in Q, && l_i w_i \le A_{seg}\ \forall i \in Q, \\
& \sum_{i=1}^{b} A_i \le \mathcal{A}_{max}, && w_i \ge w_{min}\ \forall i \in Q, \\
& \sum_{i=1}^{K} I_{j_i} + I_x = 0\ \forall j \in P \ \ \text{(KCL)}, \\
& I_i \le I_{max}.
\end{aligned}
\tag{4.45}
$$

η in the above problem is the weights for the two objectives $V_{\text{IR total}} = \sum_{i=1}^{b}|I_i|\,\frac{\rho l_i}{w_i}$, and $A = \sum_{i=1}^{b} l_i w_i$. The challenge in solving the above problem is to search for the optimal value of η at first and subsequently find the optimal value of the parameters present in the two objective functions. Even if we consider the boundary conditions, for eg:, $\eta = 0$ and $\eta = 1$ case. For $\eta = 0$ case, our objective is



(a)



(b)

Figure 4.11: IR drop profile of *ibmpg*2 circuit (a) before optimization, (b) after optimization

to minimize the $\sum_{i=1}^{b}|I_i|\,\frac{\rho l_i}{w_i}$, which is a nonlinear equation with the variables I_i and $w_i,\ \forall i \in \{1, b\}$ (ρ and l_i are constant). SLP approach is not equipped to solve nonlinear equations with large number of variables. Even if we apply SLP for a subcircuit of this objective, it fails to converge due to the nonlinear nature of the equation. Similarly, for the case $\eta = 1$, our objective function is to minimize $\sum_{i=1}^{b} l_i w_i$, with variables $w_i,\ \forall i \in \{1, b\}$. As this function is a simple proportional function of w_i, the minimum of it is achieved when the w_i is selected as a minimum, which is constrained by the other constraints mentioned in (4.45). Therefore, for the simple cases of $\eta = 0$ and $\eta = 1$, we have found some partial solutions to our problem. However, for all other cases of $\eta \in (0, 1)$, the problem consists of sum of weighted value of two objectives, which is a multimodal function. SLP cannot find any feasible solutions to this multimodal function, satisfying all the constraints. Hence, the evolutionary computing technique is one of the best approaches for such kind of complex multimodal problems. Moreover, it is also a widely used practice to solve multimodal optimization problems using evolutionary computing technique [91]. We employ the NSGA-II-based evolutionary computing approach, as it can handle multiple conflicting objectives. Although the evolutionary computing approach cannot guarantee optimality of the solution. However, we can still obtain a near-optimal feasible solution for a complex problem like the above using evolutionary computing, unlike the SLP approach, where it is challenging to obtain a feasible solution for complex problems.

From the above discussion, it is clear that the SLP approach cannot produce any feasible solutions for our problem.

4.7.2.5 Comparative Study of Proposed Work with Our Single Objective Problem

In this section, we compare results of our proposed work with our work of single objective problem formulation. The demerit of our first work mentioned in subsection 4.3 is that we have achieved IR drop minimization at the cost of increase in metal routing area. Although it would not be entirely fair to compare this work with the work mentioned in subsection 4.3, as the motivation of both works is different. However, just for the sake of getting an idea how the IR drop reduction and area are inter-dependent on each other, we list comparison in Table 4.9. In the Table, percentage change in area after optimization is denoted by ΔA. A negative value of ΔA represents a reduction in area, and a positive value of ΔA represents an increment in area. We are also comparing the worst-case IR drop reduction after optimization, and change in this quantity is represented as $\Delta V_{\text{IR worst}}$. The initial value of worst-case IR drop before the optimization is denoted as $V_{\text{IR worst initial}}$. From the Table, we can observe

that in single objective formulation, the objective is to minimize the IR drop, which is achieved at the cost of an increase in the chip's metal routing area. However, if we want a trade-off between IR drop and area, we should use multiobjective design space exploration. We obtain one of the near-optimum trade-off points for IR drop and area using the proposed framework. The evolutionary algorithm helps determine this trade-off, which also proves the effectiveness of the evolutionary approach in solving the power grid design space exploration problem. Now how much trade-off between IR drop and metal routing area is required for a certain power grid design cab be determined by the designer with the help of our proposed framework.

Table 4.9: Comparative Study of Proposed Framework with work of single-objective formulation

PG_circuits	Before optimization	**single-objective formulation (subsection 4.3)**			**This work**		
	$V_{\text{IR worst initial}}$ (V)	$V_{\text{IR worst}}$ (V)	$\Delta V_{\text{IR worst}}$	ΔA	$V_{\text{IR worst}}$ (V)	$\Delta V_{\text{IR worst}}$	ΔA
ibmpg2	0.0369	0.0263	-28.72%	+17.87%	0.0342	-7.31%	-8.51%
ibmpg3	0.2438	0.1879	-22.92%	+14.33%	0.2230	-8.53%	-8.20%
ibmpg4	0.0086	0.0041	-52.32%	+22.65%	0.0075	-12.79%	-7.92%
ibmpg5	0.0690	0.0431	-37.53%	+18.27%	0.0610	-11.59%	-7.59%
ibmpg6	0.2063	0.1529	-25.88%	+15.82%	0.1881	-8.82%	-7.32%

4.8 Conclusion

In this chapter, first we address the issue of IR drop minimization in power grid network. We observe that IR drop minimization can be obtained at the expense of increase in metal routing area. Further, a multi-objective framework for minimization of the IR drop-metal routing area of the VLSI PGN is proposed using evolutionary computation technique. Here, we also mention about the power grid design process and all reliability issues during the design phase are also included. Subsequently, the objective function considering the IR drop and metal routing area is formulated with the consideration of reliability and yield-based design constraints. NSGA-II based multi-objective evolutionary algorithm has been employed to minimize the two objectives of the problem simultaneously and to obtain an optimum point of trade-off. Before minimization, KLU solver is used to perform the power grid analysis. Current and voltage of the whole PGN are obtained from the power grid analysis, which is again fed as input to the NSGA-II optimization algorithm. Experimental results on standard IBM power grid benchmarks show that our proposed framework is able to obtain an optimum point for the IR drop and metal routing area. Hence, we can recommend the following for adaptation of evolutionary computation technique in the on-chip power grid design phase:

- Different critical design objectives can be formulated as an optimization problem for on-chip

power grid design.

- Evolutionary computation technique can be used to obtain a trade-off between different critical on-chip power grid design parameters by solving the optimization problem.
- The optimization process still takes considerable amount of time (13097 secs/3.63 hours for *ibmpg*6) for large-scale PGN. However, this gives an initial idea to the designers about selecting different critical design parameters, which, of course, make the overall design process faster.

In this chapter, we have observed heuristic approaches for power grid design optimization. In the next chapter, machine learning technique for power grid design is described.

5

Power Grid Design using Machine Learning

Contents

5.1 Introduction 76
5.2 Preliminaries and Motivation 77
5.3 Nonlinear Optimization Formulation 79
5.4 Proposed PowerPlanningDL Framework 81
5.5 Experimental Results 85
5.6 Conclusion 91

5.1 Introduction

In the last chapter, we have seen how heuristic approaches can be used for design space exploration of power grid design. In this chapter, we present how we can adapt machine learning technique for the design of power grid network of a chip.

The primary objective of the power planning phase in the backend-design of a System-on-chip (SoC) is to design a power grid network which can deliver power to all the components of the SoC within the allowed margin of IR drop and Electromigration (EM) for the durability of the chip. If these margins are not satisfied, then IR drop and EM violation can occur, which reduces the reliability of the chip. Designing a reliable power grid is an iterative process which requires many phases of incremental design to verify the power grid, as shown in Fig. 5.1. As a result of this, the design cost and power planning sign-off time increases. Therefore, to reduce the cost and the design cycle time, in this work we propose to utilize the historical data of the power planning design cycle and come up with a deep learning model which can generate a reliable power grid. Adaptation of our deep learning model in the power planning phase within the electronics design and automation (EDA) industry reduces cost and increases the efficiency of the total design phase of the chip.

In this work, we are the first to:

- Present a power planning methodology using the Deep Learning Approach in the VLSI Physical design cycle.
- We present a new aspect of obtaining a similarity between power grid design and deep learning. We also build a reliability-aware framework for power grid design using deep learning.
- We demonstrate ∼6× speedup in power grid design using the proposed framework compared to the conventional approach for power grid designs of IBM processor.

At the VLSI Physical Design level, we answer the following questions:

1) How much practically feasible Deep Learning is for the Power Planning phase?

2) How accurately can the Deep Learning approach predict different design parameters, while still satisfying the allowed IR drop and EM margin?

3) What is the efficiency of the Deep Learning approach compared to the standard power planning

Parts of the work of Chapter 5 is published in proceedings of IEEE/ACM Design, Automation and Test in Europe (DATE 2020), Grenoble, France, by S.Dey et al. "PowerPlanningDL: Reliability-Aware Framework for On-Chip Power Grid Design using Deep Learning".

tools?

These are the fundamental questions that need to be addressed for the successful adaptation of Deep learning approach in the power planning phase.

The chapter is arranged as follows. Section 5.2 contains all the necessary preliminary details and motivation of the manuscript. Section 5.3 shows the nonlinear formulation of the power grid design problem and its equivalence with deep learning training, which is used for solving the power grid design problem. Section 5.4 contains the proposed framework. The experimental results are listed in Section 5.5. The chapter is concluded in Section 5.6.

5.2 Preliminaries and Motivation

5.2.1 Fundamentals of Power Planning

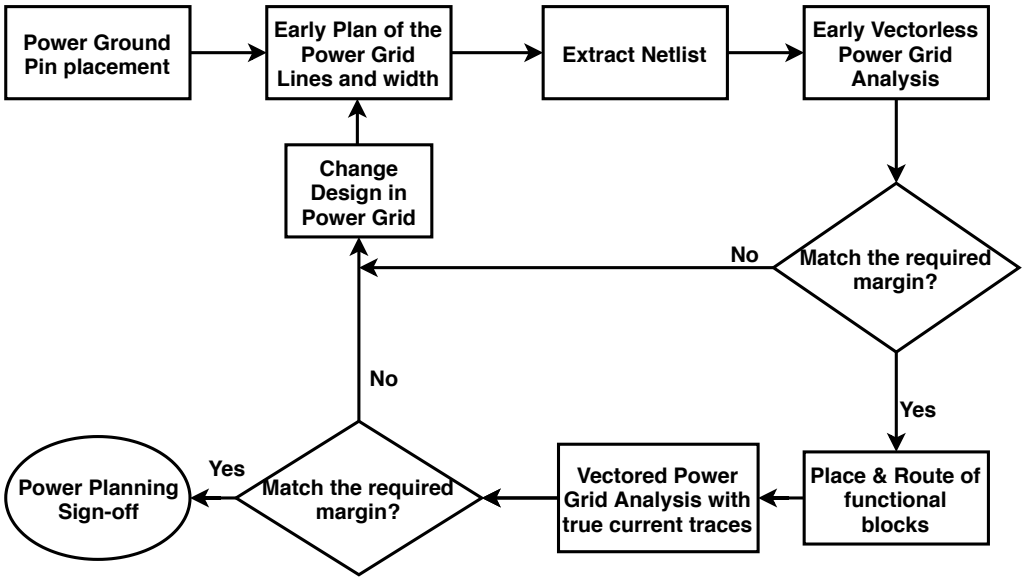


Figure 5.1: Conventional Power Planning Flow in VLSI Physical Design

Power Planning is one of the most critical stages in VLSI Physical design. The conventional power planning steps are shown in Fig. 5.1. Power planning starts with the pin placement phase of the power and ground pads. Power network is generated in order to provide power to standard cells and macros within the acceptable IR-Drop margin. Steady-state IR Drop occurs due to the resistance of the metal wires of the power grid network. IR drop can be reduced by decreasing the voltage differences between different nodes, which is determined by the power grid analysis. Early vectorless power grid analysis is done in order to find the IR drop even before the placement and routing stage with the power

information from the front end design. Once the margin of IR drop limit is satisfied in this stage, then the placement and routing are done. Subsequently, vectored power grid analysis is performed with the exact current traces of the underlying functional blocks in order to satisfy the IR drop margin. This work is a first-of-its-kind using a deep learning approach and focuses on the static IR drop and EM-aware power grid design. Therefore, this work does not consider the decoupling capacitor (decap) placement phase.

5.2.2 Related Work

5.2.2.1 Conventional Approaches in Power Grid Design

There are many works in the literature in last two decades which deal with power grid designs, analysis, optimization and verification using different heuristics. Some of the recent works on the power grids are discussed here. Fawaz et al. [44] have proposed a methodology for accurate verification of the power grids. Wang et al. [45] have proposed electromigration-aware power grid design. Heo et al. [46] have done IR drop mitigation by inserting power staple. All the methods mentioned above suffer from large convergence time.

5.2.2.2 Learning Approaches in Power Grid Design

There are very less efforts for the application of learning-based methods in power grid design. However, few closely related works are discussed here. Cui et al. [52] proposed a machine learning technique for power grid analysis by doing matrix-reordering. Fang et al. [53] proposed machine learning-based dynamic IR drop prediction. Liu et al. [54] proposed power supply noise aware circuit test timing prediction using machine learning. Chang et al. [55] in their work proposed to generate routability-driven power grid network using machine learning techniques. Lin et al. [56] proposed IR drop prediction of ECO-revised using machine learning. Ye et al. [57] proposed the voltage droop mitigation using support vector deep. Cao et al. [58] proposed a learning-based method to predict the quality of power grid network package. There is not much significant work in the literature on the deep learning-based power planning methodology.

5.2.3 Motivation

Designing a power grid is similar to solving a non-linear optimization problem, which is proved in the next section. Similarly, training deep neural networks is considered as solving a nonlinear optimization problem. Therefore, we try to investigate the underlying similarity between the two problems and try

to solve the power grid design problem using deep learning. Apart from that, deep learning has been successful in predicting complex tasks in many areas of science and technology. Therefore, we use deep learning for prediction of the power grid design, which reduces the design cycle time and dependence of human intervention for the initial design of the power grid.

5.2.4 Overview of the Proposed Methodology

Our objective here is to reduce the iterative flow of the power planning phase while still satisfying the allowed margin of IR drop and Electromigration with the help of the historical data generated in the design process of the power grid network. Therefore, initially, we perform the feature extraction and prepare the training data using these historical data of the design phase and specifications, as shown in Fig. 5.2. Subsequently, we train our deep learning model using these historical data and predict a power grid design for any new design specifications.

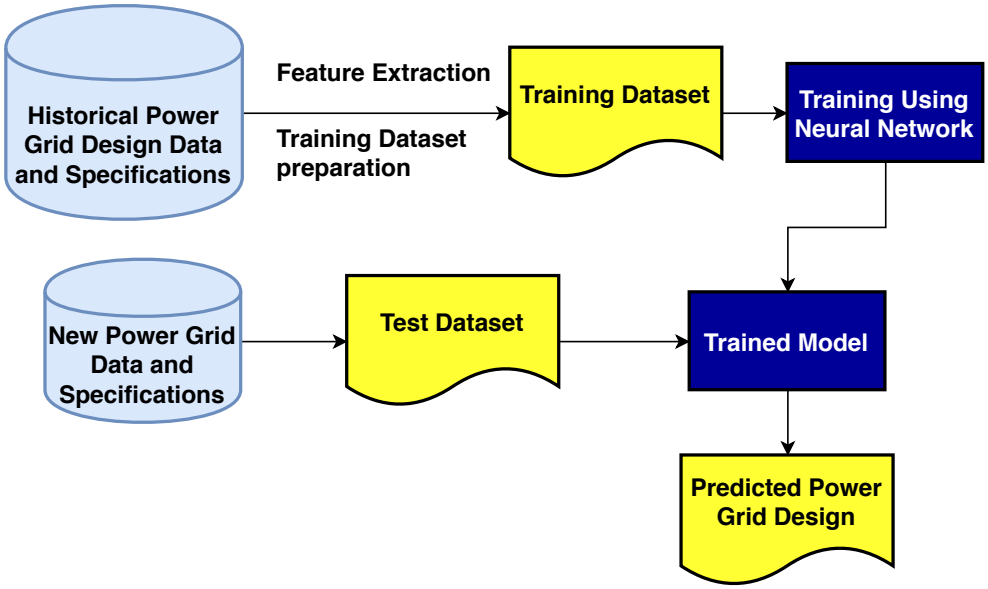


Figure 5.2: Proposed Deep Learning-based Power Planning Flow

5.3 Nonlinear Optimization Formulation

In this section, we prove the equivalence between power grid design and deep learning. The objective of the power grid design is to obtain the optimum width of the power grid lines considering different reliability constraints. If the IR drop across the i^{th} power grid lines (V_{IR_i}) is represented as $V_{IR_i} = I_i R_i$, where $R_i = \rho \frac{l_i}{w_i}$, ρ is sheet resistance, l_i = length, w_i = width, I_i = current through it. From here we

can write that,

$$w_i = \rho \frac{l_i I_i}{V_{IR_i}}, \tag{5.1}$$

which is nonlinear function with variables l_i and V_{IR_i} (considering I_i to be constant). It is also well-known from [92] that training of a deep neural network is also a nonlinear optimization problem. Therefore, both the power grid design and training of a deep neural network are similar. Using this comparison, we build the neural network model for the power grid design problem which is shown in Fig. 5.3 Then the minimization of the power grid design objective function can be represented as

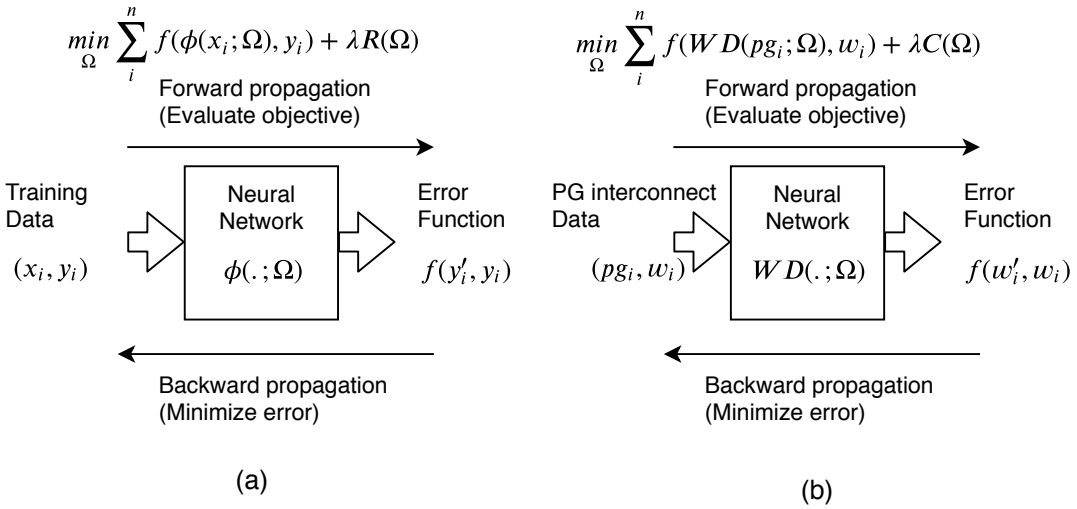


Figure 5.3: Equivalence between deep neural network training and power grid design (a) Training a neural network for weights Ω (b) Solving power grid design with neural networks for weights Ω.

follows,

$$\min_{\Omega} \sum_{i}^{n} f(WD(pg_i;\Omega), w_i)) + \lambda C(\Omega), \tag{5.2}$$

where pg_i is the each instance of the power grid interconnect, $WD()$ is the cost function of (5.1) as predicted by the neural networks for weights Ω. $f()$ is the error function or loss function to evaluate the error form the true value. $C()$ is the reliability and other constraints of the power grid design which are described below, and can be satisfied using the weight λ.

The relation between width of the power grid lines (w_i) and the spacing between the two power grid lines (s_i) can be represented as follows

$$\sum_{i=1}^{K} s_i + w_i = W_{core}, \tag{5.3}$$

where W_{core} represents the ring width. For large number of power grid lines, designing power grids with such constraints mentioned in (5.1) and (5.3) become difficult and tedious process. The EM reliability constraint for maximum current density J_{max} can be defined as,

$$\frac{I_i}{w_i} \leq J_{max}. \tag{5.4}$$

These constraints need to be satisfied while designing the power grid using neural network, which are denoted as $C()$ in (5.2), can be adjusted with weight λ.

5.4 Proposed PowerPlanningDL Framework

5.4.1 Problem Formulation

A floorplan of an SoC with the power grid lines and underlying functional blocks is shown in Fig. 5.5(a). While designing the power grid, it is very challenging to predict the optimum widths of the power grid lines. Over-designing the power grid lines by increasing the power grid line widths increase the total metal routing area of the chip. If it is under-design in order to reduce the metal routing area, then the power grid suffers from unwanted IR drop and Electromigration effects due to the increase in resistance and current density of the metal lines. Simultaneously, the design rules need to be taken care of while over-designing/under-designing. The correct predictions of the widths of the power grid lines can reduce different iterations of the power planning phase. Therefore, in our deep learning adaptation, we use a supervised learning approach to create a model. Our model learns the optimum widths of the metal lines from previous historical data which are obtained for IR drop and Electromigration resistant power grid designs with some allowed margin. Subsequently, we use this learned deep learning model in order to predict the widths of power grid lines for a new design.

As shown in (5.1), w_i is dependent on V_{IR_i} and I_i, which can only be found after power grid analysis. As power grid analysis is time-consuming, we want to evade the power grid analysis phase. Therefore, we are using an alternate approach to predict w_i. We are using X-coordinate, Y-coordinate (of the planned floorplan of the underlying functional blocks), and its switching current activity (I_d) (which is obtained from the from front-end phase in value change dump (VCD) file) to predict w_i. The reason for choosing these as features are shown in Section 5.4.2. Considering this we have formulated two problems to be solved given as follows,

Problem 5.1. *Given an X-coordinate, Y-coordinate of floorplan and the switching activity of the current (I_d) for that point, then predict the metal width required for that location which can satisfy the*

IR drop and EM constraints. (Please refer to Fig 5.4)

Problem 5.2. *Given the width and the switching activity of the PG interconnects, predict the IR drop of the PG interconnect.*

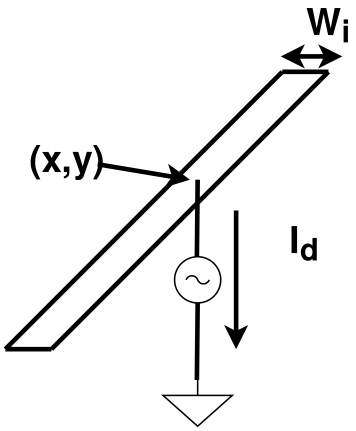
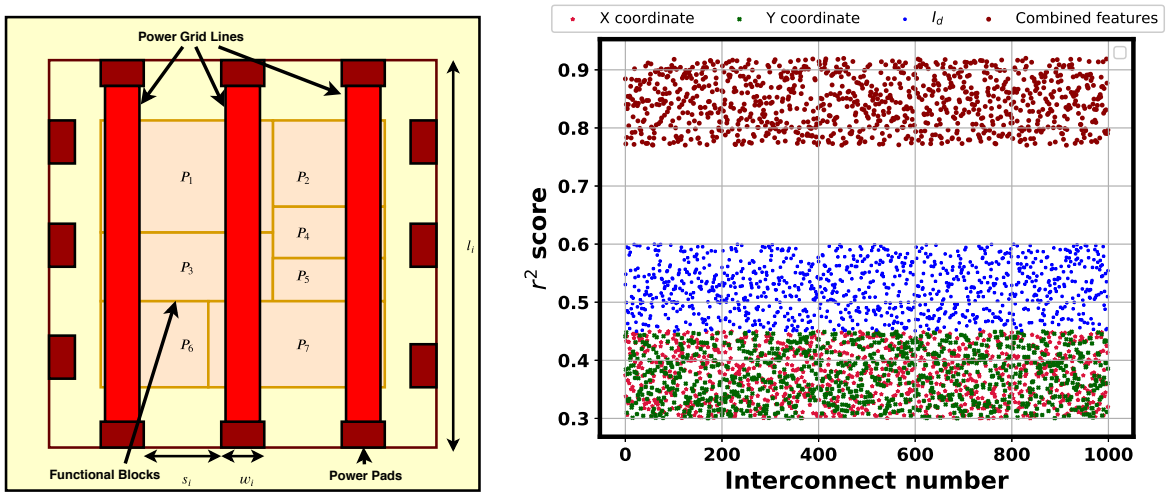


Figure 5.4: Metal line showing its features (x, y) position and I_d switching current and width w_i.



(a) (b)

Figure 5.5: (a) A floorplan of an SoC with the power grid lines over the functional blocks. (b) Variation of r^2 scores for 1000 power grid interconnects of *ibmpg*1 benchmark circuit with different input features

We are using a multi-target regression technique to model the deep learning model where we consider multiple input features (independent variables, pg_i) as the input to our model and numerous output features (dependent variables, w_i). Mathematically, it can be represented as

$$\text{Predict } w_i \ \forall i \in \chi, \tag{5.5}$$

where $\chi = \{pg_i\}$ for all $i \in \{1, 2, \cdots, n\}$ power grid interconnects is the training dataset.

5.4.2 Feature Selection & Training Data Preparation

Definition 5.1. *(r^2 score) or coefficient of determination is a metric which shows the goodness of the prediction for the regression method. A value closer to($\leq$) 1 is desired for the data to fit in the model properly.*

For selecting various features for our deep learning model, we evaluated the r^2 score of different input features with the w_i. It has been observed that the combination of the input features X-coordinate, Y-coordinate (of the planned floorplan of the underlying functional blocks), and its switching current activity (I_d) fits to be the best for the neural network-based multi-regression technique as it has higher r^2 score(Please refer, Fig. 5.5(b) and Table 5.1).

Table 5.1: r^2 score of different input features and output feature w for a PG interconnect.

Input Features	X coordinate	Y coordinate	I_d	Combined
r^2 score	0.34	0.39	0.61	0.89

I_d is the current obtained from the switching activity of the functional blocks having (X,Y) coordinate. Therefore, the training dataset is generated with the quadruple (X coordinate, Y coordinate, I_d, w_i) from some of the real power grid designs.

5.4.3 Neural Network-based Deep Learning Model

The neural network has one input, one output, and hidden layers. An illustrative example is shown in Fig. 5.6. There can be many number of hidden layers. We have used 10 hidden layers in our model, which is obtained by hyperparameter optimization. This neural network is trained with quadruple

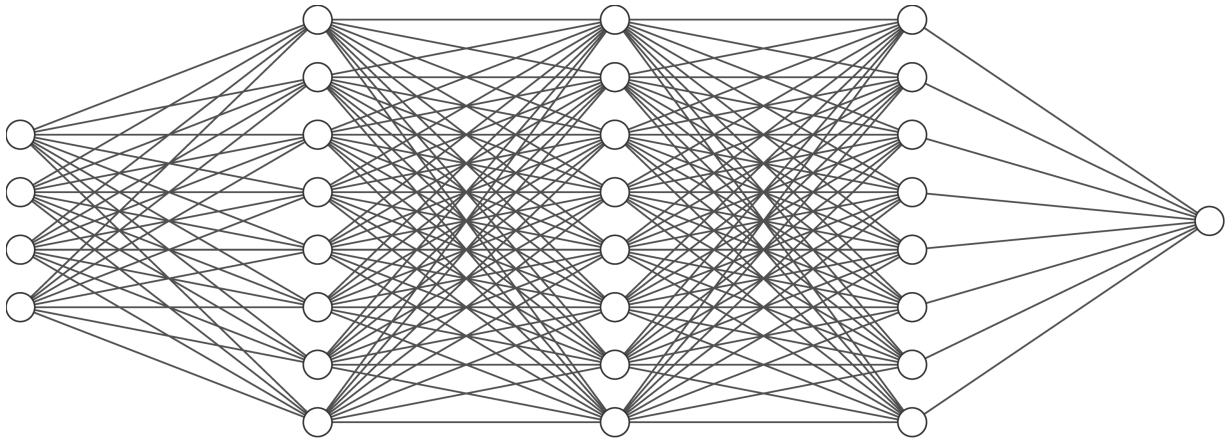


Figure 5.6: A sample neural network with three hidden layers.

(X coordinate, Y coordinate, I_d, w_i) for different weights Ω as part of its forward propagation step

as mentioned in Section 5.3. Subsequently, Adam optimizer [93] is used to minimize the loss or error function as a part in the backpropagation step. Once trained, the new test samples can be used to predict w_i.

5.4.3.1 The Power Grid Interconnect Width Prediction

The power grid interconnect width prediction is given below in Algorithm 5.1.

Algorithm 5.1: Wire width prediction by NN

Input: Training Set
Output: w_i and gradient

```
ForwardPropagation(X coordinate, Y coordinate, I_d, w_i)
{
    return loss function f
}
BackwardPropagation()
{
    return gradient
}
```

5.4.3.2 IR Drop Prediction

The IR drop prediction algorithm is given below in Algorithm 5.2. From Algorithm 5.1 after testing

Algorithm 5.2: IR drop prediction

Input: Predicted width w_i, I_d
Output: Predicted IR drop

1 From switching current I_d and w_i;
2 Use Kirchoff's law to predict IR drop.

on test dataset, we already have the w_i, which means we have the R_i of the power grid interconnect (considering l_i to be constant). We need the I_i to find the IR drop across the interconnect. The following approach helps in obtaining I_i. The number of power grid lines which are required can be obtained using the following formula:

$$\#PG\ line = \frac{W_{core}}{w_i}, \tag{5.6}$$

where W_{core} represents maximum width of the core. As shown in the Fig. 5.5(a), if we consider that i^{th} power grid line carry, I_i current. Then the current requirement of each of power grid lines to the

blocks can be represented as follows,

$$I_1 = I_{11} + I_{13} + I_{16} \tag{5.7}$$

$$I_2 = I_{21} + I_{23} + I_{27} \tag{5.8}$$

$$I_3 = I_{32} + I_{34} + I_{35} + I_{37}, \tag{5.9}$$

where I_{ij} represents current provided by i^{th} power grid line to the j^{th} block. From the above, we can obtain current through the interconnect and subsequently the IR drop.

5.4.4 Test Data Generation

Test dataset is generated by perturbing the same dataset which is used for training. The perturbation is done by changing the branch current, node voltage, and switching current of the underlying functional blocks by a $\gamma = 10\%$, which is termed as perturbation size. Experiments are done in the next section by varying the perturbation size in order to see the variation in prediction accuracy.

5.4.5 Post-refinement Stage

In order to obtain the optimal hyperparameters and unbiased evaluation of the test dataset for the learning model, a validation set is created in this post-refinement stage. This step is more of a frequent evaluation of the learning model in order to ensure high-level predictability on unseen samples without violating IR drop and EM reliability constraints. For the validation set, a similar type of dataset as used in test dataset is used. With this post-refinement phase, the objective is to reduce the prediction error on a new set of unseen data and correspondingly obtain the hyperparameters, in order to avoid overfitting.

5.5 Experimental Results

5.5.1 Simulation Setup

The framework is developed with *C++* and *python*. For deep learning operations *Tensorflow* library of the python has been used on a Linux machine with Intel Xeon E5-2650 processor, with the GPU configuration Nvidia Tesla K20c. The datasets are generated, and the proposed PowerPlanningDL is validated using the IBM Power Grid benchmarks [3], which are standard power grid benchmarks extracted from IBM processors. The details of the IBM PG benchmarks are listed in Appendix A. Current loads of the IBM PG benchmarks are modified in order to obtain the desired effects. The

simulation setup for the experiments is set according to Fig. 5.7. All the hyperparameters of the neural network are fixed for which the best results are obtained. The hyperparameters are listed in Table 5.2.

Table 5.2: Hyperparameters used in the learning framework

Items	Value
#Hidden layers	10
Activation	ReLU
Optimizer	Adam
Learning rate	0.001
β_1	0.9
β_2	0.999
ϵ	1e-08

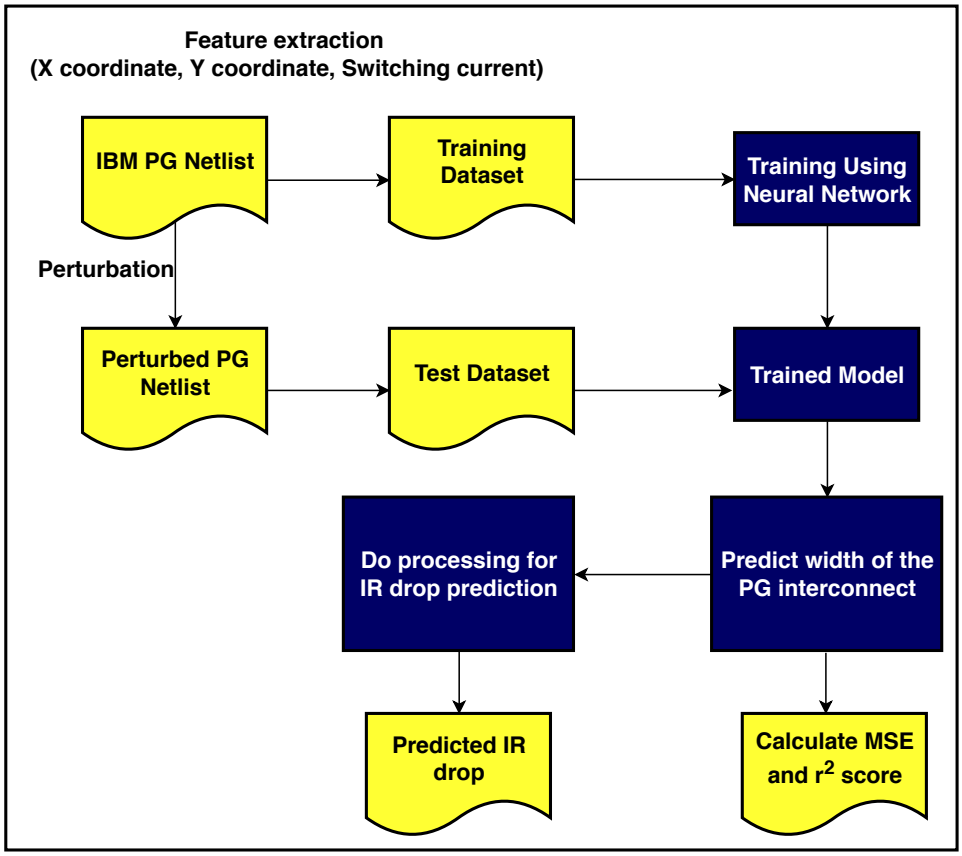
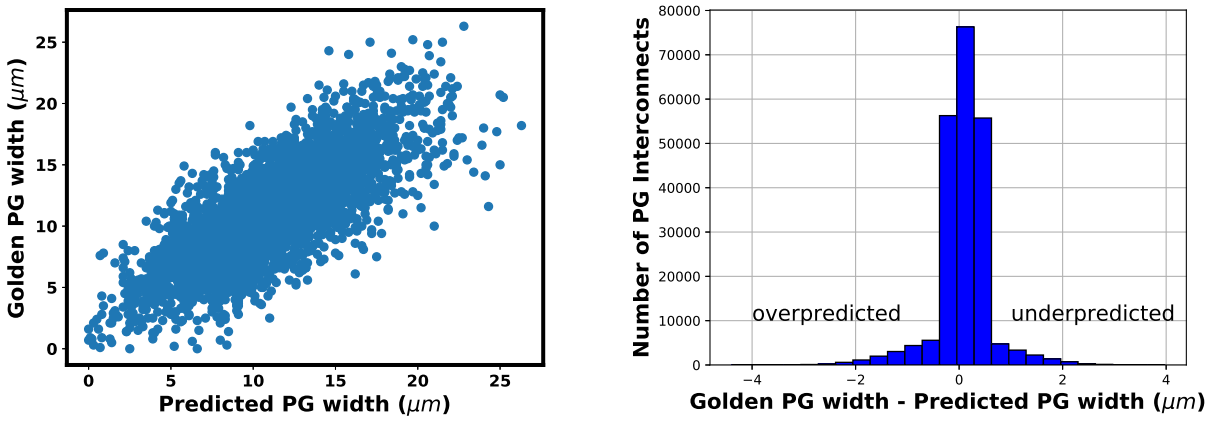


Figure 5.7: Flow of the simulation setup of the Deep Learning Flow

5.5.2 Study of Predicted Power Grid Interconnect Width

In this section, the correlation between the predicted width of the power grid using PowerPlanningDL and conventional approach evaluated. From the correlation value, it can be seen how much

the predicted widths are related to the golden width obtained from the conventional approach. The correlation plot is shown in Fig. 5.8(a). To study the error distribution of the predicted widths, the error histogram plot is shown in Fig. 5.8(b) (Horizontal axis represents error). From the error histogram, we can observe that most of the predicted widths are concentrated near 0, meaning most of the predicted widths of PG interconnect produce near about 0 error. As the amount of error increases, the number of power grid instances decreases. From this result, we can conclude that the predicted widths of the power grid lines using PowerPlanningDL are very close to the golden results generated by the conventional approach for most of the interconnects.



(a) (b)

Figure 5.8: Power Grid interconnect width prediction for *ibmpg2* benchmark circuit (a) Correlation scatter plot (b) Error histogram (Horizontal axis represent the error).

5.5.3 Study of Predicted IR Drop in Power Grid

The IR drop map is plotted for the conventional approach and also for the PowerPlanningDL approach, as shown in Fig. 5.9 for *ibmpg2* circuit and *ibmpg6* circuit. The worst-case IR drop for all the benchmarks are listed in Table 5.3. From the IR drop map and the worst-case IR drop values, it can be inferred that the PowerPlanningDL can predict the IR drop close to the conventional approach.

5.5.4 Main Result: Study of Convergence Time

The convergence time for both the approach is shown in Table 5.4. Convergence time of the conventional approach includes the IR drop analysis time, as it is the primary time-consuming task.

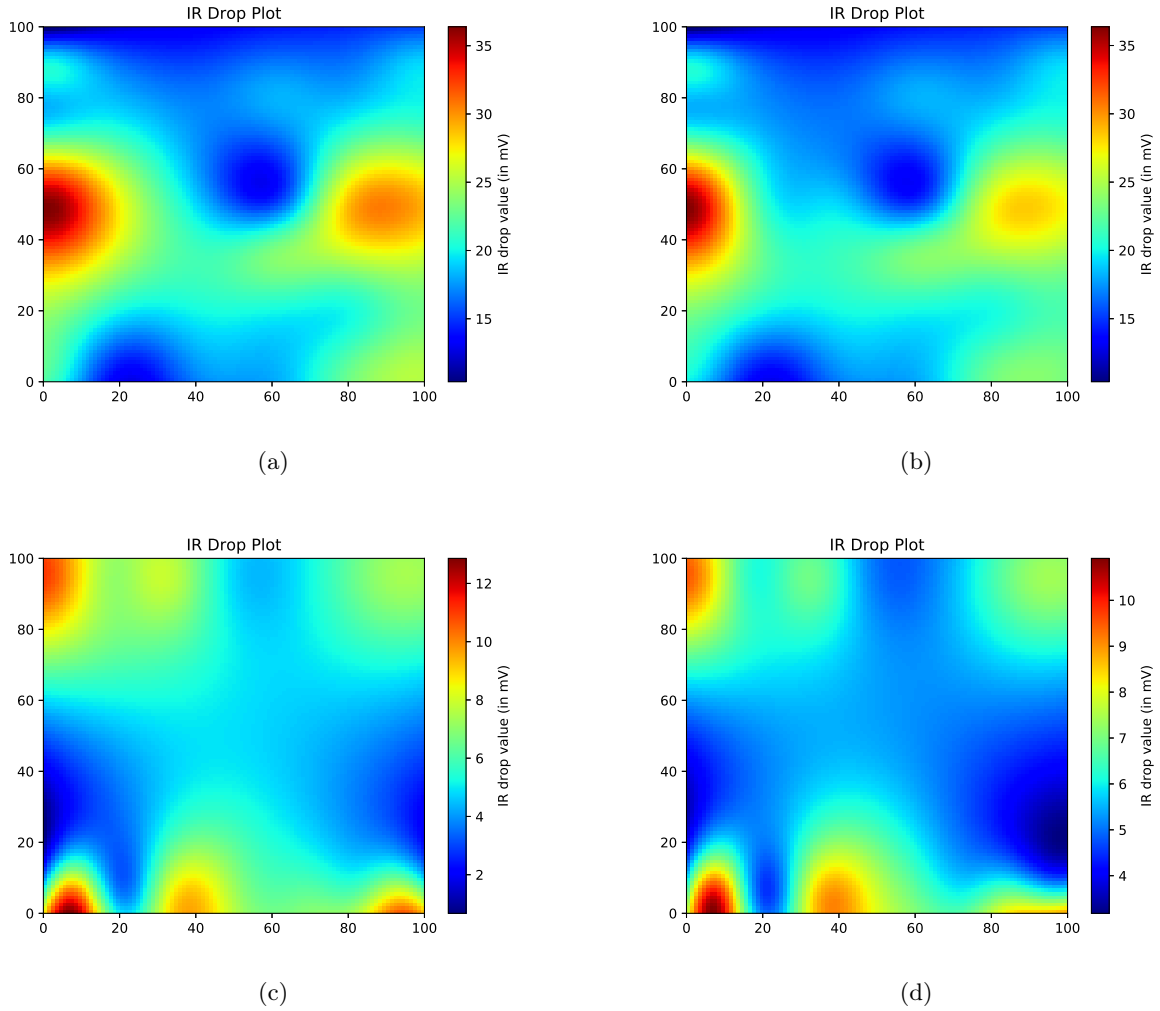


Figure 5.9: IR drop map of (a) Conventional method *ibmpg*2 circuit (b) PowerPlanningDL methodology *ibmpg*2 circuit (c) Conventional method *ibmpg*6 circuit, and (d) PowerPlanningDL methodology *ibmpg*6 circuit.

Table 5.3: Comparision of Worst-case IR drop using Conventional power planning approach and PowerPlanningDL framework

	Worst-case IR drop (mV)	
PG circuits	**Conventional**	**PowerPlanningDL**
*ibmpg*1	69.8	68.2
*ibmpg*2	36.3	36.1
*ibmpg*3	18.1	18.0
*ibmpg*4	4.0	4.1
*ibmpg*5	4.3	4.2
*ibmpg*6	13.1	13.0

For the PowerPlanningDL, the convergence time shows the prediction time of the width and IR drop prediction time, as mentioned in Section 5.4. From the table, it can be seen that our proposed PowerPlanningDL is 5.87× faster than the conventional approach for the *ibmpg*5 benchmark. It is also observed that for larger benchmarks the speedup is more, as larger grids take more time for power grid analysis in conventional approach, which is not used in our PowerPlanningDL framework. That is one of the main reasons that we get a significant speedup for our PowerPlanningDL compared to the conventional approach. We achieve the speedup at the cost of accuracy. It is to be noted that for the convergence time of the conventional approach reported in Table 5.4, we have considered the best-case scenario and reported the convergence time only for one iteration of the design cycle. In the worst case, there can be multiple iterations of the design cycle, for which the conventional approach takes much more time, whereas the convergence time remains the same for PowerPlanningDL in all scenarios. This also shows the advantage of PowerPlanningDL in reducing the number of iterations in the design cycle.

Table 5.4: Comparison of convergence time for Conventional power planning approach and PowerPlanningDL framework

	Time (sec)		**Speedup**
PG circuits	**Conventional**	**PowerPlanningDL**	$\frac{\textbf{Time}_{\textbf{Conventional}}}{\textbf{Time}_{\textbf{PowerPlanningDL}}}$
*ibmpg*1	6.85	3.56	1.92×
*ibmpg*2	23.46	11.88	1.97×
*ibmpg*3	29.50	8.07	3.59×
*ibmpg*4	52.4	11.83	4.42×
*ibmpg*5	74.80	12.74	5.87×
*ibmpg*6	97.5	17.41	5.60×
*ibmpgnew*1	102.58	21.50	4.77×
*ibmpgnew*2	48.60	10.86	4.47×

5.5.5 Overhead: Study of Model Accuracy

The mean square error (MSE) can be defined as,

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - y_i')^2, \tag{5.10}$$

where y_i represents the actual width and y_i' represents predicted width. The r^2 score, and MSE using the proposed framework is listed in Table 5.5. MSE tells about the prediction error (overhead of deep learning approach) while predicting the interconnect width. From this result of MSE, we can conclude that the proposed PowerPlanningDL can predict the power grid design, which is very close to the golden design generated by the conventional approach. From r^2 score we know how well the data is fit

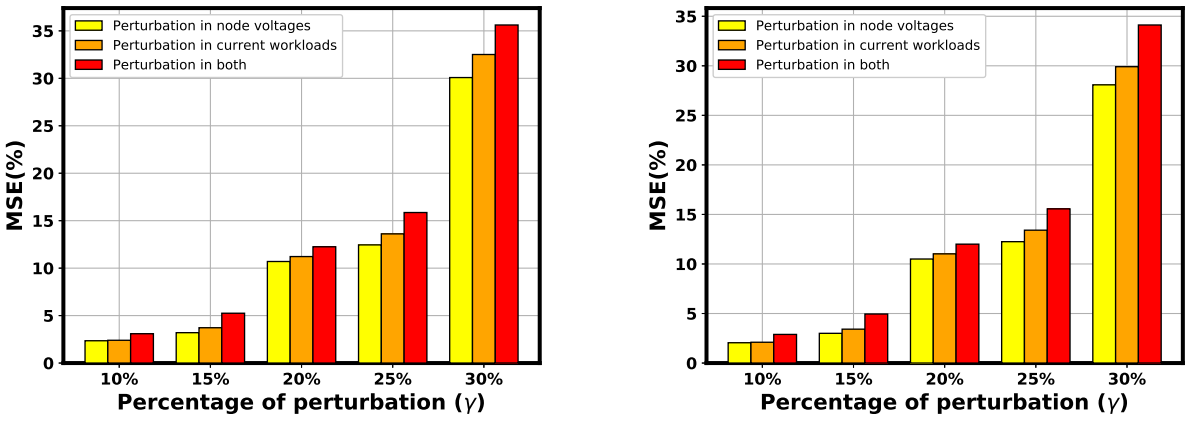
in the model.

Table 5.5: r^2 score, MSE and Peak memory using PowerPlanningDL framework for all the IBM PG benchmarks. Please note 1 Gigabyte = 953 Mebibyte (Mib).

PG Circuits	#interconnects	r^2 score	MSE	Peak Memory (in MiB)
*ibmpg*1	30027	0.933	0.0231	66
*ibmpg*2	208325	0.937	0.0230	318
*ibmpg*3	1401572	0.932	0.0212	730
*ibmpg*4	1560645	0.941	0.0210	749
*ibmpg*5	1076848	0.944	0.0225	511
*ibmpg*6	1649002	0.945	0.0208	841
*ibmpgnew*1	2352355	0.943	0.0201	1025
*ibmpgnew*2	1422830	0.945	0.0209	745

5.5.6 Study of Variation of MSE with Perturbation Size

The variation of MSE with the perturbation size ($\gamma\%$) is shown in Fig. 5.10. It is observed that as the perturbation size increases the MSE increases. From this observation, we can infer that the proposed PowerPlanningDL is best suited for the incremental-based power grid design, where we need to generate the power grid for little changes (or perturbations) in the design.



(a) (b)

Figure 5.10: Comparison of prediction accuracy on test set in MSE with variations in perturbations size for (a) *ibmpg*2 (b) *ibmpg*6 benchmark circuit.

5.5.7 Study of Peak Memory

For the completeness of the results, we have also evaluated the memory profile of the proposed framework using the *mprof* tool. The memory profile of the proposed framework for two benchmark

circuits *ibmpg*2 and *ibmpg*6 are shown in Figure 5.11. We also show the peak memory usage for all the IBM PG benchmarks as listed in Table 5.5.

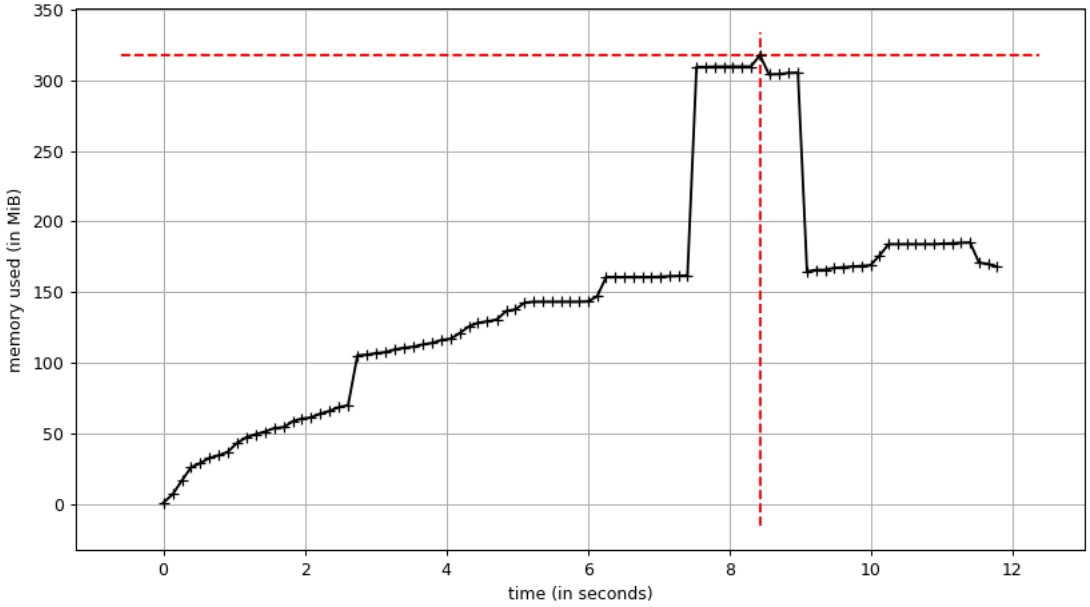
5.6 Conclusion

In this chapter, we have proposed a deep learning-based framework PowerPlanningDL to predict the initial power grid design. For the first time, we have shown the equivalence between the neural network training and power grid design. We predict the power grid interconnect width as part of the design process, which is time-consuming and tedious work. Subsequently, we also anticipate the worst-case IR drop in the power grid. A neural network-based multi-regression technique is used in our model for accomplishing the prediction tasks. Results on IBM power grid benchmarks show ∼6× speedup than the conventional power grid design approach. We have also performed various other experiments.

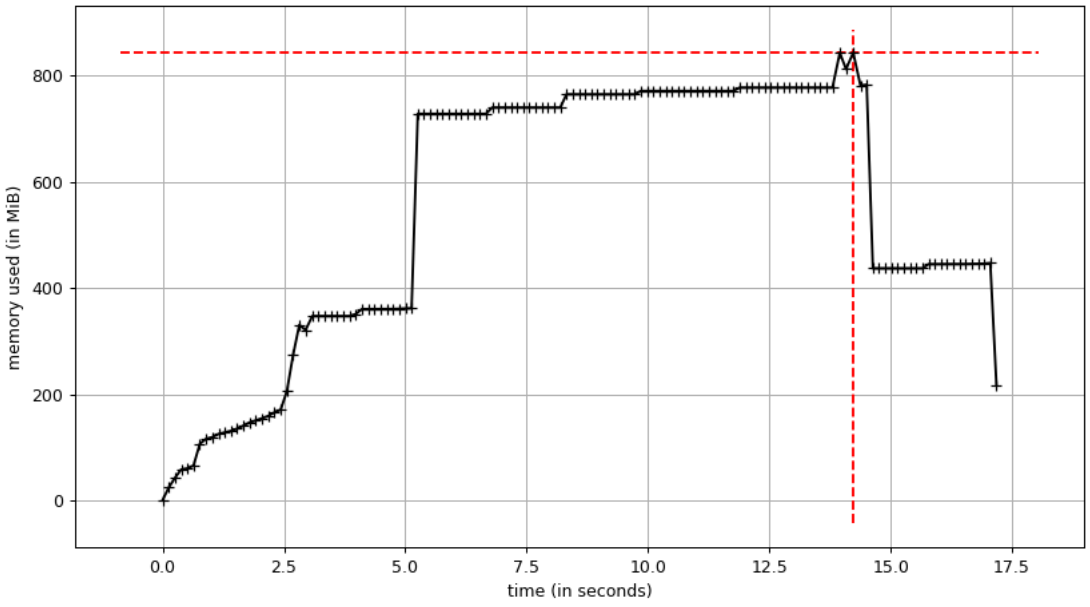
From the results of the experiments, we can recommend the following for the adaptation of the deep learning in power planning phase of VLSI Physical Design:

- The predictability of the deep learning approach is close to the conventional method, with very less convergence time (∼6× speedup).
- Deep learning in power planning is useful in the incremental-based power grid designs, where the perturbation size is small.
- The error due to the prediction increases for the PowerPlanningDL framework for the designs with large perturbations.
- Finally, from this work, we can say that the industry can adapt the deep learning approach for the power grid design, which reduces many iterative steps in order to obtain an appropriate initial design.

In the next chapter, we present an adaptation of a machine learning approach for electromigration-aware aging prediction of on-chip power grid network.



(a)



(b)

Figure 5.11: Memory used by PowerPlanningDL for (a) *ibmpg2* benchmark circuit and (b) *ibmpg6* benchmark circuit. 1 Gigabyte (GB) = 953.674 Mebibyte (MiB)

6

Aging Prediction of Power Grid Design using Machine Learning

Contents

6.1 Introduction 94
6.2 Backgrounds 97
6.3 Proposed Machine Learning Model 100
6.4 EM Assessment using ML Model 107
6.5 Experimental Results 112
6.6 Conclusion 126

6.1 Introduction

In the last chapter, we have observed how machine learning techniques can be used for the design of on-chip power grid. In this chapter, we demonstrate application of machine learning technique in the aging prediction of on-chip power grid network.

Electromigration (EM) has become one of the major reliability issues for the interconnects of the newer technology nodes [8, 94, 95]. As metal interconnects suffer most due to the EM-based reliability issues, the power grid (PG) network of a VLSI Chip is highly susceptible to EM, activated by the momentum transfer of free electrons to the metal atoms. While the signal and clock line also suffer from EM degradation, these lines carry bidirectional current and which results in a longer lifetime due to the so-called *healing* effects. However, power grid interconnects mostly carry a unidirectional current which has no privilege of healing effect and as a result are more susceptible to EM degradation.

The standard practice in the industry is to use traditional Black's model [13] for all the interconnects of the PG network and considering the earliest branch failure time as the lifetime of the entire grid which takes hours of time. For successful EM sign off, EM violations of the interconnects need to be checked iteratively by changing the design incrementally in every step. Therefore, to speedup the EM sign off process, incremental analysis of the EM is necessary. Recently many works on physics-based EM models are proposed for optimistic aging prediction of the PG network. Huang et al. [4] have proposed such a physics-based model for the first time. Mishra et al. [47] proposed a better approach for predicting the lifetime of the PG network considering transient stress modeling. Chatterjee et al. [5] proposed a fast physics-based electromigration assessment using an efficient solution of linear time-invariant systems. Wang et al. [45] proposed a physics-based model using integral transformation technique. Chatterjee et al. [6] extended their work on LTI system-based EM assessment approach by incorporating macro-modeling-based filtering and predictor approach . There are several other works on physics-based EM assessment model [48–50]. However, most of these methods still take hours of time to obtain the lifetime of the chip as it involves solving partial differential equations (PDE). Practical adaptation of these physics-based methods [4–6, 45, 48–51] for a full scale EM prediction for a chip is not possible as these methods are time-consuming. Although, recent work of Najm and Sukharev [7] shows a significant speedup for full chip simulation in EM-aging prediction by employing Monte-Carlo

Parts of the work of Chapter 6 is published in ACM Transactions on Design Automation of Electronic Systems (TODAES), by S.Dey et al. "Machine Learning Approach for Fast Electromigration Aware Aging Prediction in Incremental Design of Large Scale On-Chip Power Grid Network".

Simulation. However, the work of [7] still requires a fresh EM-simulation for incremental changes in the design. Therefore, in order to speed up the EM-sign off phase of the incremental PG network design and to facilitate a practical method which can be adapted for full scale EM prediction, it is better to reuse the historical data (generated by the standard aging models) and use these data to create a machine learning model which can instantly predict the EM-aging of the PG network.

Motivation of Machine Learning Approach for EM-Aging Prediction: Recently, EM-aging has emerged as the design problem [96]. The traditional practice of determining EM-based aging is time-consuming. If there is an incremental change or any small perturbations in the power grid design during its design phase, then to check for the acceptable EM margin once more takes hours of time using conventional EM-aging computation models. As machine learning (ML) has been proved to be an effective approach for predicting tasks over the last few decades, this approach can be used to predict the EM-aging of the PG network, which would take very little time to obtain EM-aging of the PG network compared to the traditional approaches. In literature, there are few developments of ML-based approach. Huang et al. [97] used a machine learning approach in detection and classification of defects in TSV-based 3D IC. Elfadel et al. [98] have reviewed many applications of VLSI CAD using Machine Learning. In our machine learning approach, the model has to be trained just once for a certain power grid topology and subsequently it can predict the EM-aging for small perturbations in the PG network with very little testing time (CPU runtime). Therefore, in this Chapter, using a machine learning approach we predict the EM-aging of the PG network of a chip by constructing a regression-based model. Our method uses the historical data generated from the traditional time-consuming EM-aging computation model to learn the EM-aging of the PG Network and eventually able to predict the EM aging time. Our main motivation for this chapter is to demonstrate that the EM-aging prediction of the PG network can be done using Machine learning approach. Experiments on different PG benchmarks showed that using the historical data of EM-aging model for our Machine Learning approach reduces the overall EM sign-off time significantly in comparison to all the state-of-the-art models [4–7] and the predicted MTTF is also found to be better than that of [5–7] and comparable to [4].

Novel contribution: To the best of our knowledge, this work is the first to use the Machine Learning approach to predict the EM-aging in incremental design of the VLSI PG network. The major contributions of our proposed work are:

- Fast KLU solver is used for power grid analysis, from which currents and voltages of the power grid networks are obtained. These values are utilized in feature extraction and training data generation phase.
- Feature selection is made evaluating the r^2 score of different features. Accordingly, the training dataset is generated using the power grid benchmarks, and the Black's model is used for labeling the training data. Test data set (which is different from the training data set) is generated by perturbing the same benchmarks data.
- EM-aging prediction problem of the VLSI PG network is formulated as a regression-based supervised machine learning model by selecting different features (related to the properties of EM).
- Different regression-based Machine Learning models have been used to obtain the best model for EM-aging prediction by evaluating the model accuracy metrics and other performance metrics.
- A new failure criterion has been proposed analytically based on the worst-case IR drop of the PG network, which is utilized with the machine learning model to predict the EM-aging of the PG network.
- The proposed approach for EM-aging prediction using the machine learning model can predict the EM-aging for the test data set. The accuracy of the EM-aging model is demonstrated by changing different test data sets with a variation of perturbation size.
- Further, a logistic-regression based classification model is used to obtain the potentially weak EM affected metal segments of the PG network.
- Experimental results on different power grid benchmarks show that the proposed EM-aging prediction approach using ML is faster than all the state-of-the-art models [4–7]. The predicted value of MTTF is also found to be better than that of [5–7] and comparable to [4]. This proves the efficiency of the proposed ML-based in the EM-aging prediction model in the incremental design of the VLSI power grid network.

The rest of the chapter is organized as follows. In Section 6.2, EM fundamentals and PG network model used in this work have been discussed. Section 6.3 contains the problem formulation for the aging prediction, training data generation, and the proposed machine learning model is described. Section

6.4 describes the utilization of the ML approach for EM-aging prediction. This section also describes a new failure criterion of the PG network. Identification of the potentially EM-affected degraded metal segment using a logistic regression-based classification technique is explained in this section. Different experiments on the standard power grid benchmarks are carried out to validate our proposed method using machine learning in Section 6.5. The chapter is concluded in Section 6.6.

6.2 Backgrounds

6.2.1 Electromigration fundamentals

Electromigration is the process of movement of metal atoms due to the exchange of momentum from the electrons to the metal atoms. The EM degradation can happen in two phases: *void nucleation* and *void growth*. Under high current in the metal lines, metal atoms are subjected to stress for a prolonged period of time, which causes the void to occur. This EM degradation phase is termed as *void nucleation* phase. Once the void nucleates, it started to grow which is termed as *void growth* phase. The aging of the metal lines due to EM degradation is measured as *mean-time-to-failure* (MTTF). Black [13] has proposed an empirical equation to evaluate the MTTF of the metal interconnects due EM, which is given as follows,

$$MTTF = \frac{A}{J^n} e^{\frac{E_a}{kT}}, \tag{6.1}$$

which evaluates the interconnect MTTF based on known current density (J) and temperature (T). A is a constant depends on the metal geometry, grain size, and current density. The value of n is found to be 2 which, E_a is the EM activation energy, and k is the Boltzman's constant. Blech [14] observed that the mortality of the metal interconnects vary with the length. He proposed a criterion for the filtration of immortal interconnects, which follows,

$$(JL) \leq (JL)_c = \frac{\Omega \sigma_c}{eZ\rho} \tag{6.2}$$

L is the length of the metal interconnect, Ω is the atomic volume, e is the electron charge, eZ is the effective charge of the migrating atoms, ρ is the resistivity of the metal interconnect, σ_c is the critical stress needed for the failure of the metal interconnect. Korhonen et al. [15] proposed a mathematical formulation to represent the *hydrostatic stress* σ which originates from the influence of EM.

$$\frac{\partial \sigma}{\partial t} = \frac{\partial}{\partial x}\left[c\left(\frac{\partial \sigma}{\partial x} + \frac{eZ\rho J}{\Omega}\right)\right], \tag{6.3}$$

where $c = \frac{D_a B \Omega}{kT}$, where D_a is the atomic diffusivity, B is the bulk modulus. Approximate value of the nucleation time can be obtained from (6.3). A void nucleates once the stress exceeds the critical value.

Physics-based Models: The basic idea of the physics-based models is solving the interconnect trees, which are generated from the power grid netlist. Initially, the current densities of all the interconnect trees are calculated. Subsequently, PDEs associated with the interconnects (refer (6.3)) are solved in order to find the stress level. The solution of the PDE gives us the nucleation time (the time required for nucleation) for critical values of stress. This nucleation time dominates the MTTF value. In [4], the authors use an iterative method that looks after the changes in stress and the power grid resistance as a function of time. With time the stress level, as well as the resistance of the power grid network, exceeds a critical value, and the simulation stops. The cumulative nucleation time is considered as MTTF of the power grid network. Power grid analysis is done in every iteration to observe the voltage drop level (V_{ir}) of the interconnects, which acts as the stopping criteria of the simulation. If the voltage drop level reaches above a threshold level (V_{th}), the power grid is considered dysfunctional, and the lifetime is calculated. This process is described pictorially in Fig. 6.1. As it is an iterative process, it takes a large amount of computational time to converge. Therefore, there is a need to have new methods for EM aging evaluation, which can converge very fast.

6.2.2 Power Grid Model

EM is a long-term phenomenon which considers the average effects of the currents, therefore, a DC load model of the PG network (Fig. 6.2) with only resistive elements of the metal line is considered [99]. From the steady-state model of the PG network, the system of linear equations is obtained which can be written as $\mathbf{GV} = \mathbf{I}$, where $\mathbf{G}$ is the conductance matrix, $\mathbf{V}$ is the node voltages vector, and $\mathbf{I}$ is the current sources vector connected to grounds. Even though the probabilistic solvers [17] and locality-based solvers [100, 101] have gained a lot of attention in the recent past. Direct solvers have always been the first choice for solving the system of linear equations. In this work, we use KLU-based direct solver [12] to obtain the node voltages vector $\mathbf{V}$. Furthermore, using all the node voltages of the PG network, branch currents can be obtained. The hydrostatic stress of all the interconnects can be obtained by using (6.3) under a given applied current condition. Now if we perform integration of

Interconnect trees: connected graph of the interconnects.
Nucleation: first stage of formation of a void.

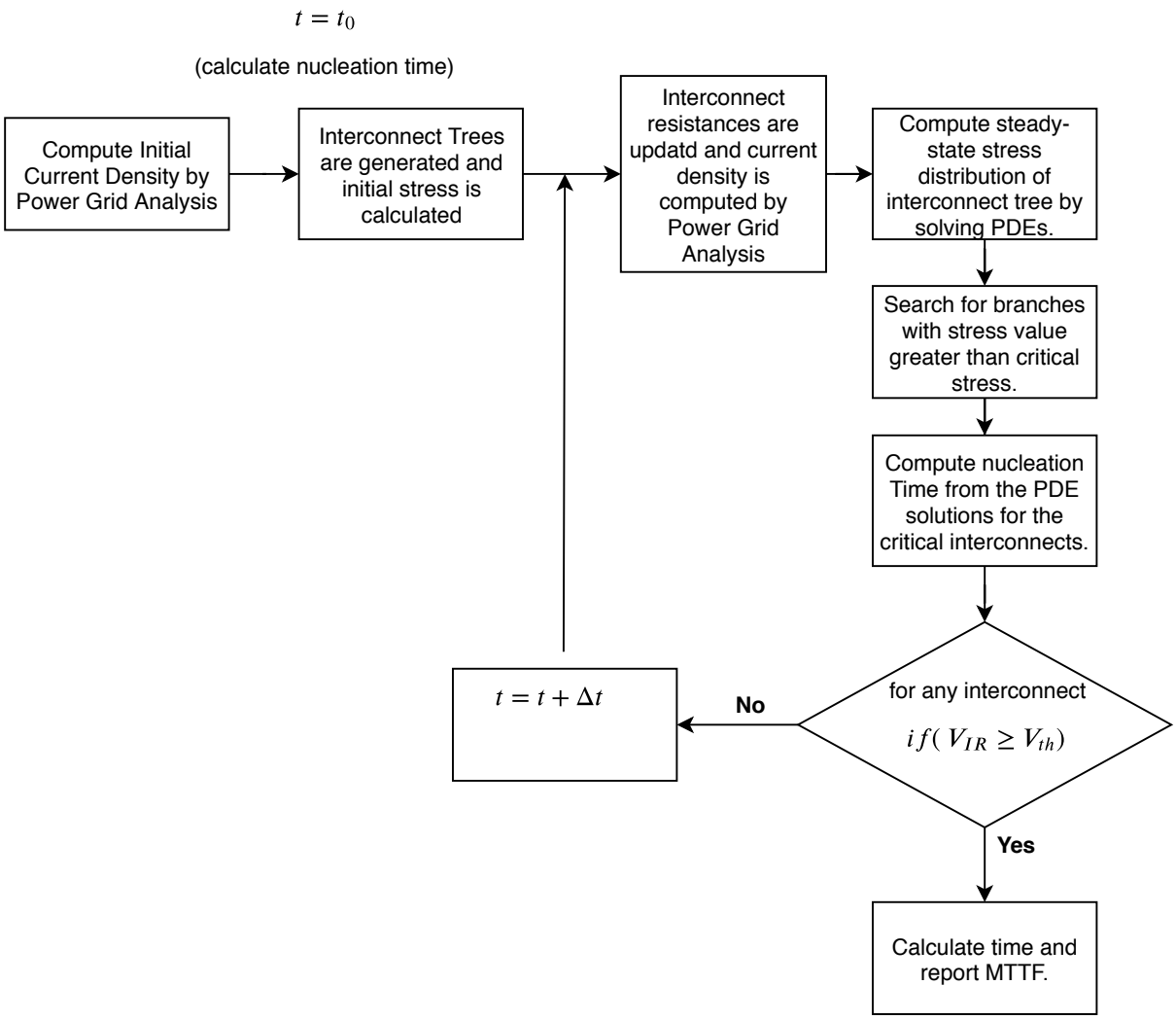
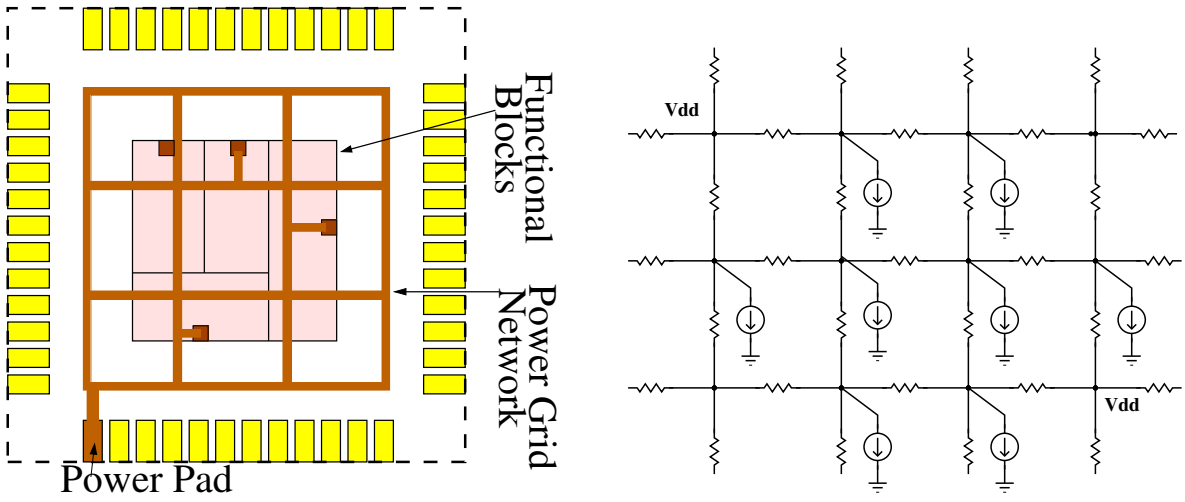


Figure 6.1: Basic Idea of Physics-based EM model



(a) (b)

Figure 6.2: (a) Floorplan of a VLSI SoC with its PG network connecting the functional blocks. (b) The resistive DC load model of the PG network.

(6.3), we obtain the expression of σ:

$$\sigma(t,x) = \sigma_0 + \frac{\partial}{\partial x}\left[\int_0^t c\frac{\partial \sigma}{\partial x}dt + c\frac{eZ\rho}{\Omega}\int_0^t Jdt\right] \tag{6.4}$$

which describes that the hydrostatic stress distribution of an interconnect depends on the time integral of the applied current density. This can be considered as a justification of substituting the current waveforms with the time averaged DC current. Therefore, effective current density of the interconnect metal lines can be represented by [102],

$$J = \frac{1}{T}\left(\int_0^T J^+(t)dt - \psi\int_0^T |J^-(t)|dt\right), \tag{6.5}$$

where ψ is the EM recovery factor and found experimentally, $J^+(t)$ is the current density from one side of the wave. For the unidirectional current of the PG network, the effective-current density is the time-averaged current density.

6.3 Proposed Machine Learning Model

6.3.1 Problem Formulation

The flow of our supervised machine learning model for EM-aging prediction is shown in Fig. 6.3. The input to our supervised machine learning model for EM-aging prediction is a set of n training samples of interconnects of a PG network, where each training sample i has m *input features* (denoted

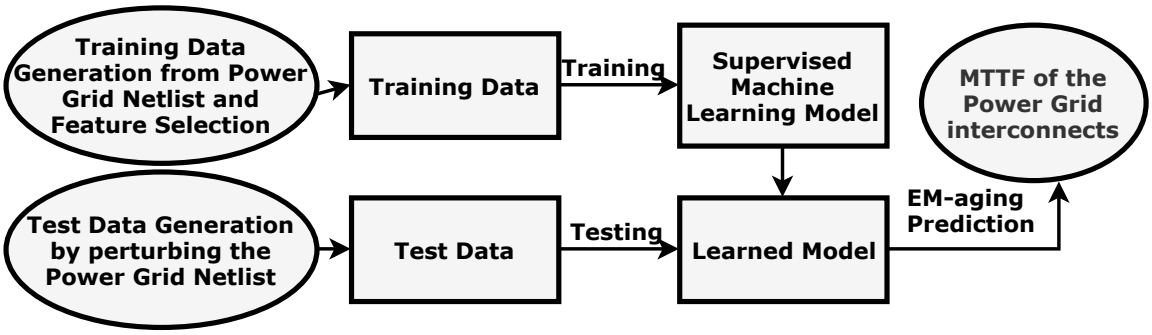


Figure 6.3: Flow of Machine Learning Model for EM-aging Prediction

as X_i) and one *output feature* (denoted as y_i). The objective is to train our model with the input features and to predict the output of a new test sample τ (denoted as y'_τ) by a function of only the input features. For our aging prediction model, the output feature is the MTTF of the interconnects of the PG network. The input features are related to the properties of EM for different interconnects of the PG network. Regression is the process of constructing the relationship between independent variables (input features) and dependent variables (output features) in order to estimate the output of the dependent variables. In the simple regression model, we only consider a single input feature for prediction. To make the prediction correct, we can consider different input features which is considered as multiple regression. In our proposed machine learning model, we have considered a few input features of the interconnect of the PG network to demonstrate the effectiveness of the EM-aging prediction model. We have used the neural network as our core machine learning technique since this technique is proved to be the most effective for the supervised learning problem.

6.3.2 Feature Selection and Training Data Generation

As shown in Fig. 6.4, we know that the MTTF depends on many parameters. However, it has been observed that MTTF changes significantly with the variation of current density (J), temperature (T), and the length of the interconnect (L). Hence current density, temperature, and interconnect length can be considered as the input features of our aging prediction machine learning model. In Chapter 5, we show that variation of IR drop of the metal lines results in a variation of MTTF of the PG network, henceforth, IR drop of all the interconnects can also be considered as an input feature. Therefore, current density, length of the interconnect and IR drop of the interconnects considered as input features for our machine learning model for aging prediction. The coefficient of determination (r^2 score) [103] between the MTTF and different input features is listed in Table 6.1. r^2 score denotes

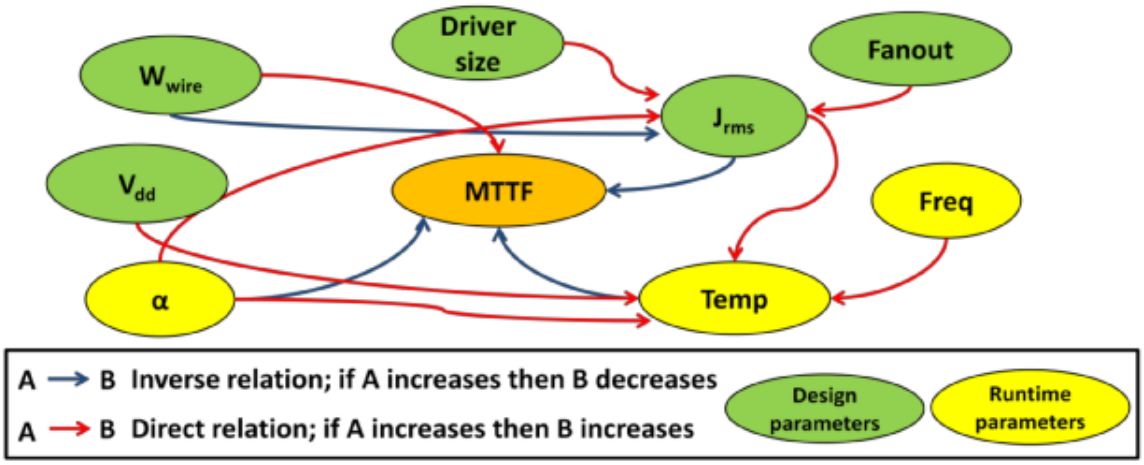


Figure 6.4: Design and Runtime parameters affecting EM MTTF [2].

the proportion of variance of the MTTF (output feature) that is predictable from the input features. A higher value of $r^2 (\leq 1)$ is desired for selecting the best combination of input features for the model. This is the main intention of using r^2 score for feature selection so that we know which model fits properly for our EM-aging dataset. The r^2 score of Table 6.1 demonstrates that the combination of J, L, IR drop, and T as the input features (as multiple regression) result in a more accurate regression model which predicts MTTF value nearest to the actual value. For the purpose of completeness, we are defining the r^2 score definition for our EM-aging prediction model as given in Definition 6.1.

Definition 6.1. *(r^2 **score**) The coefficient of determination (or r^2 score) between the MTTF (output feature) and different input features denote the proportion of variance of the MTTF (output feature) that is predictable from the input features is defined as follows,*

$$r^2 = 1 - \frac{SS_{residual}}{SS_{total}}, \tag{6.6}$$

where $SS_{residual} = \sum_i \left(y_{\tau_i} - y'_{\tau_i}\right)^2$ is residual sum of squares and $SS_{total} = \sum_i \left(y_{\tau_i} - \bar{y_\tau}\right)^2$ is total sum of squares. Here, y_{τ_i} denotes actual value of the i^{th} test sample, y'_{τ_i} denotes predicted value of i^{th} test sample, and $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_{\tau_i}$. A larger value of $r^2 (\leq 1)$ is desirable for the data of the input features and output feature to perfectly fit in the regression model.

Proposition 6.1. *The supervised machine learning model for EM-aging prediction fits best as multiple regression model with the combination of input features J, L, IR drop and T.*

Proof. From the Definition 6.1, we know that a higher value of r^2 score is desirable for the best fit of the data for the EM-aging model. A simple experiment to find r^2 score with a different combination of input features show that for the combination of J, L, IR drop and T, the r^2 score is highest as listed in the Table 6.1. Therefore, we can say that our EM-aging model fits best for the combination of input features J, L, IR drop and T. □

Proposition 6.1 describes the feature selection for our EM-aging model. More about the EM-aging

Table 6.1: r^2 score between MTTF vs different input features

Input features	J	L	IR drop	T	J, L, T, and IR drop combined
r^2 score	0.638	0.753	0.859	0.781	0.978

model property with different input features can be found in the ablation study of the model. The training data containing the input features and output feature is generated using the Algorithm 6.1 for different power grid benchmarks. Initially, power grid analysis is done using KLU solver [12] to obtain all the branch currents of the PG network. Subsequently, IR drop of all the interconnect is obtained and finally using the Black's [13] series method MTTF for all the interconnects is obtained. For finding the temperature (T) of the PG interconnects, we have used the thermal model as proposed in [104]. In this way, for a PG network, we have generated the training set which is used to train the regression model. The flow of the machine learning model for EM-aging prediction is shown in Fig. 6.3. We briefly discussed IR drop analysis using KLU Solver, as it is an important part of the training data generation phase.

IR drop analysis using KLU Solver: At first, the power grid circuit in the form of the SPICE netlist is feed as input to the EM aging prediction framework. IR drop analysis is done in order to obtain the currents and voltages of the power grid circuit. Subsequently, using the modified nodal analysis(MNA), the system of linear equations is solved, which gives us the currents and the voltages of the power grid circuit. KLU solver [12] is employed for solving the system of linear equations, as it has a more considerable speedup than the HSPICE circuit simulator [90], which is demonstrated in Table 6.5. The better speedup of KLU is obtained, as it efficiently determines the solution of the linear system of equations resulting from the modified network analysis. The matrix is permuted in block triangular form in KLU solver, and each block is ordered to reduce the fill. Gilbert-Peierls algorithm is employed for performing the LU factorization, and the system is subsequently solved using block back substitution [12]. Once the power grid circuit is solved using the KLU Solver, and all circuit parameters such as current and voltages are determined. These parameters are used for obtaining the training dataset of the power grid circuit.

6.3.3 Proposed ML Model using Neural Network Regression

Neural network is a nonlinear function which takes some input features and gives some output feature as shown in Fig.6.5. These input and output layers of a neural network form an arrangement of

Algorithm 6.1: Training Data Generation

Input: Power grid netlist.
Output: Training dataset.
1 Interconnect length (L) is extracted from netlist;
2 Power grid analysis is done using KLU solver to find the currents of all the interconnects.;
3 J is calculated for the interconnects;
4 IR drop of each of the interconnect is obtained;
5 Temperature (T) of each of the interconnect is calculated using the thermal model [104];
6 Using Black's model, MTTF of each of the interconnects of the PG network is obtained;

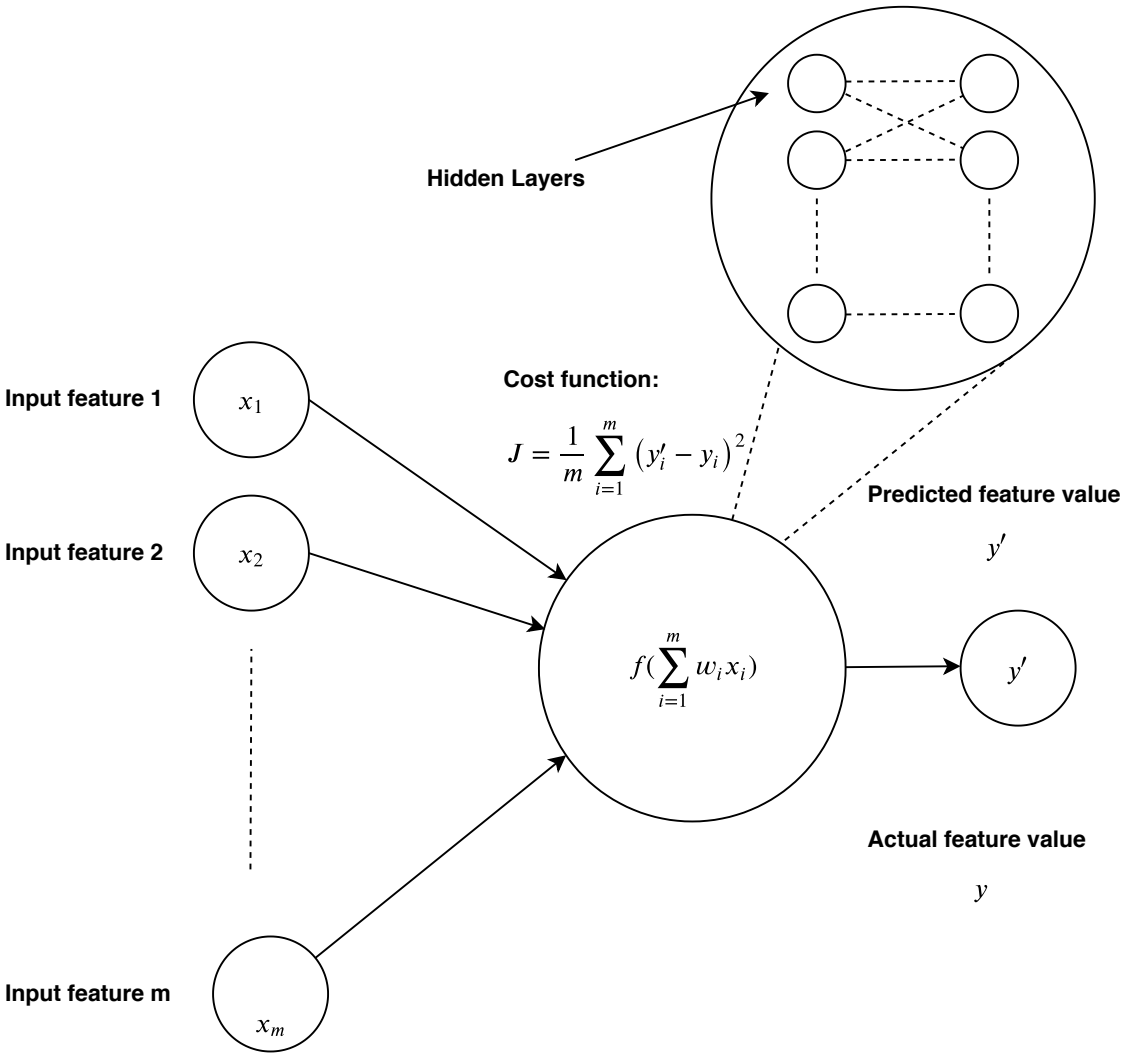


Figure 6.5: An illustration of neural network with its input features and output feature with magnified view of the hidden layers.

connected layers, where each layer contains a few neurons represented by nodes. There are three types of layers in a neural network i.e., one input layer, one output layer, and a number of hidden layers. The number of neurons in the input layer depends on the number of input features of the training dataset. For solving the regression problem, the number of neurons in the output layer is only one. There can be various numbers of the hidden layers and the number of neurons on each of the hidden layers can vary. The main aim of the neural network is to optimize the weights of each of the connections of the neurons in order to reduce the error cost function J_e.

For our EM-aging model, we considered four neurons in the input layer as we have considered four input features. Each of these four features have n training samples. We apply bias in each of the layers in order to generalize the model. We apply activation function in all the nodes which is represented as $a_u^{[l]}$ refers to the activation function of the u^{th} neuron unit in the layer l, where for the input layer $l = 0$ for simplicity of the representation. Activation functions are important, as the characteristics possessed by it is the assumed to be the characteristic of the neuron. For each neuron node, one activation function is applied. The four input features of the EM aging prediction model can be represented as follows using the unifying formulated:

$$x_1 = a_1^{[0]} \tag{6.7}$$

$$x_2 = a_2^{[0]} \tag{6.8}$$

$$x_3 = a_3^{[0]} \tag{6.9}$$

$$x_4 = a_4^{[0]}, \tag{6.10}$$

The output of the hidden layers are given as follows as mentioned in [105],

$$z_u^{[l]} = {W_u^{[l]}}^T x + b_u^{[l]} \tag{6.11}$$

$$a_u^{[l]} = g(z_u^{[l]}) \tag{6.12}$$

where, l is the l^{th} hidden unit number, and u is the u^{th} neuron in the l^{th} hidden unit. Here b represents the bias applied in each layer, W represents the weight matrix, and z represents the combination of b and W, which is further given as input to the activation function. The output of the final output layer

is given as follows:

$$z_u^{[l]} = W_u^{[l]^T} a^{[l-1]} + b_u^{[l]} \tag{6.13}$$

$$a_u^{[l]} = g(z_u^{[l]}) \tag{6.14}$$

We have used rectilinear unit (ReLU) activation function, whose response is as the following:

$$g(z) = \max(z, 0), \tag{6.15}$$

where $g(z)$ is a nonlinear function. Using the activation of the output layer i.e., (6.14) we can get the value of the MTTF predicted by Neural Network. Once the predicted value is achieved, cost function is formulated in order to increase prediction accuracy. We have used the mean squared cost function which is given below,

$$\text{Minimize} \quad J_e = \frac{1}{m} \sum_{i=1}^{m} \left(y_i' - y_i\right)^2 \tag{6.16}$$

Using any optimizer such as Adam optimizer or gradient descent approach, the cost function is optimized and the weights of the neural network are updated. In this way, accurate values of the output feature is obtained.

6.3.4 Test Data Generation and Incremental Analysis

For the purpose of test data generation we have perturbed a region of the power grid network. Perturbation is done using three ways:

1. By varying the node voltages in a region: A region with γ_1% nodes of the PG network are considered for perturbation and the voltages of each of the γ_1% nodes are changed by an amount of 1% of V_{dd}.

2. By varying the current workload in a region: Similarly, a region with γ_2% current sources of the PG network are considered for perturbation and the current values of each of the γ_2% current sources are changed by an amount of 1% of the maximum current of the PG network.

3. By varying both the node voltages and current workload in a region: In this case, a region with γ_1% of nodes of the PG network and γ_2% current sources of the PG network are considered for perturbation. Voltages of each of the γ_1% nodes are changed by an amount of 1% of V_{dd} and the current values of each of the γ_2% current sources are changed by an amount of 1% of the maximum current of the PG network.

Perturbed power grid network is solved in a faster way using the incremental power grid analysis method as done by Boghrati et al. [19]. We have adapted our KLU solver for the incremental power grid analysis to speed up the test data generation process. The adaptation comes as follows,

$$\begin{aligned}(G+\Delta G)(V+\Delta V) &= (I+\Delta I)\\ \Rightarrow (G+\Delta G)\Delta V &= \Delta I - \Delta G V\\ \Rightarrow G_{eff}\Delta V &= \Delta I_{eff}\end{aligned} \tag{6.17}$$

where $G_{eff} = (G + \Delta G)$ and $I_{eff} = \Delta I - \Delta G V$. As dimensions of ΔV and ΔI_{eff} are very small. Hence, by solving (6.17), we get the perturbed values of the power grid network instantly. We don't require to perform a fresh power grid analysis again which is time consuming. In this way the test dataset is generated for validating our EM-aging ML model.

6.4 EM Assessment using ML Model

6.4.1 EM-Aware Aging Prediction using Proposed ML Model

Algorithm 6.2: MTTF Prediction using Proposed ML Model

Input: Training dataset.
Output: MTTF of the entire PG network.
1 Train the model for Aging prediction;
2 **if** $JL \geq (JL)_c$ **then**
3 Predict MTTF using EM-aware aging prediction model for the interconnects of the test data set;
4 MTTF of the first η% mortal interconnect is considered as MTTF of the PG network as proposed in Theorem 6.1, 6.2.

Machine Learning based aging prediction model has been illustrated in Algorithm 6.2. Initially, the training data set is generated for different power grid benchmarks using the Algorithm 6.1. As the numerical value of different features varies significantly in magnitude, therefore, normalization of all the features is done in order to fit the training data properly into the regression model. Subsequently, the aging prediction model is constructed using Neural Network-based regression technique. Activation is applied in all the nodes of the Neural Network and the cost function using mean-squared-error is constructed. Adam optimizer is used to minimize the cost function. Accurately predicted values are obtained for the output feature MTTF. Therefore, for any given training features (X_i, y_i) and also the test features (X_τ, y'_τ) a nonlinear function $f(\sum_{i=1}^{m} w_i x_i)$ can be formed in terms of the weights of the connections of the different neurons (w_i) and the input features(x_i) with the help of the activation

function. In this way, once the regression model is trained using a large number of samples of training data (X_i, y_i), then the model is able to predict y'_τ for any new unseen sample X_τ. The prediction accuracy of the EM-model is evaluated using the metric mean-square-error (MSE) as described next.

Mean Square Error (MSE) is defined as the average squared difference between the estimated values using EM-aging ML model and the true values of the MTTF (output features) of all the interconnect samples of the PG network. A value of MSE closer to and greater than zero is desirable for the model to have high accuracy.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_{\tau_i} - y'_{\tau_i})^2 \tag{6.18}$$

y_{τ_i} is the actual value of the output feature of i^{th} sample of the test dataset and y'_{τ_i} is the predicted value of the output feature of i^{th} sample of the test dataset.

We obtain the predicted MTTF value from the EM-aging ML model. However, in order to achieve an MTTF value comparable to the state-of-the-art results, we propose a new failure criterion which is described next.

6.4.2 Proposed Failure Criterion For The PG Network

Chatterjee et al. [99] showed that mortality of first interconnect does not ensure the aging of the PG network as there must be some alternative current carrying path to the mortal affected line. Huang et al. [4] in their work proposed that the failure criterion for the PG Network depends on the worst case IR drop noise of the PG Network. We have analytically obtained a direct relation between the worst-case IR drop noise, and expected number of mortal interconnects as established in Theorem 6.1. From which it can be deduced that the PG Network becomes dysfunctional while the expected number of mortal interconnects is $\eta\%$ of the total interconnects which is more clearly elaborated in the Theorem 6.2. Our aim is also to obtain an optimistic prediction of MTTF of the PG Network. Therefore, in order to obtain an optimistic MTTF prediction of the PG network, we have used this new failure criterion of the PG Network. With the help of EM-aging prediction model, MTTF of the first $\eta\%$ interconnects which become mortal is calculated and is considered as the MTTF of the PG network as given in the Algorithm 6.2.

Theorem 6.1. *The worst case IR drop noise of the PG Network (caused by change in resistance due to EM) goes above V_{th}, if and only if the expected number of mortal interconnects due to EM is approximately $\eta\%$ of the total interconnects.*

Proof. Let the total number of interconnects be M. Let the total number of nodes be N. Let we have $q(\leq N)$ instances of the worst-case IR drop node. Let $K_x(\leq M)$ be the total number of interconnect

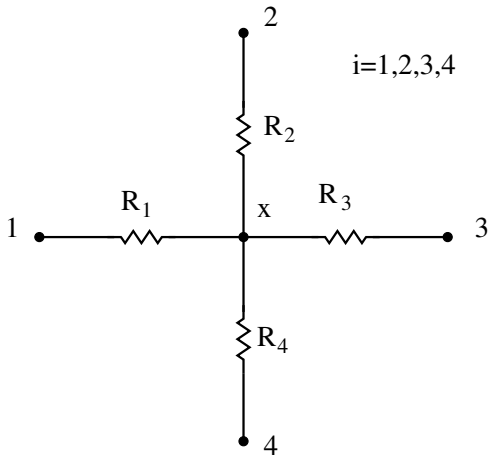


Figure 6.6: Nodes directly connected to the worst case IR drop noise node x

directly connected to the node having the worst-case IR drop noise x. If the IR drop noise across any interconnect (V_{IR}) connected to node x goes above a certain threshold voltage V_{th} (i.e., $V_{IR} \geq V_{th}$) then we assume that the interconnect fails or becomes mortal. The probability of the i^{th} interconnect (connected to node x) to fail, $P_i(V_{IR} \geq V_{th}) = \frac{v_x - v_i}{\sum_{i=1}^{K}(v_x - v_i)}$, where v_x and v_i are node voltages of the node x and i respectively (refer Fig. 6.6). The probability of any interconnect to be connected with a worst-case IR drop noise $= q.\frac{\binom{N}{1}}{\binom{N}{2}}$. The probability of the i^{th} interconnect to be connected with a worst-case IR drop noise and also to fail$= q.\frac{\binom{N}{1}}{\binom{N}{2}} \sum_{x=1}^{q} \frac{v_x - v_i}{\sum_{i=1}^{K_x}(v_x - v_i)}$. The estimated total number of interconnects (β) which are connected with a worst-case IR drop noise and also to fail is,

$$\beta = M.q.\frac{\binom{N}{1}}{\binom{N}{2}} \sum_{x=1}^{q} \frac{v_x - v_i}{\sum_{i=1}^{K_x}(v_x - v_i)}. \tag{6.19}$$

$$\Rightarrow \beta = \eta\% \text{ of } M \tag{6.20}$$

where

$$\eta = 100.q.\frac{\binom{N}{1}}{\binom{N}{2}} \sum_{x=1}^{q} \frac{v_x - v_i}{\sum_{i=1}^{K_x}(v_x - v_i)} \tag{6.21}$$

Therefore, the expected number of mortal interconnects due to EM is $\eta\%$ of total interconnects. Similarly, if we assume that total mortal interconnects is $\eta\%$ of M, then it can be obtained that the worst case IR drop noise of the PG network goes above V_{th}. □

Theorem 6.2. *The PG Network is considered to be dysfunctional if the worst-case IR drop noise goes above V_{th} [106], which is equivalent to the fact that PG Network can also be considered to be dysfunctional if the expected number of mortal interconnects is approximately $\eta\%$ of the total interconnects.*

Proof. From the Theorem 6.1, it can be concluded that the worst case IR drop noise of the PG Network directly depends on the expected number of mortal interconnects of the PG Network. The failure criterion as proposed in [106] states that the PG Network becomes dysfunctional once the worst case IR drop noise goes above V_{th}. Hence, from the Theorem 6.1, it can be deduced that the PG Network becomes dysfunctional while the expected number of mortal interconnects is approximately $\eta\%$ of the total interconnects. □

We have shown an example of a sample PG network in Example 6.1, in order to see a practical value of expected number of mortal interconnects as depicted in the Theorem 6.1.

Example 6.1. For a PG network with 9 nodes and 12 edges as shown in Fig. 6.7, the expected number of mortal interconnect is $\eta \approx 14$ as found by using the Theorem 6.1. $V_{dd} = 1.8V$, All resistors have 0.5Ω and all current source values are $0.05A$.

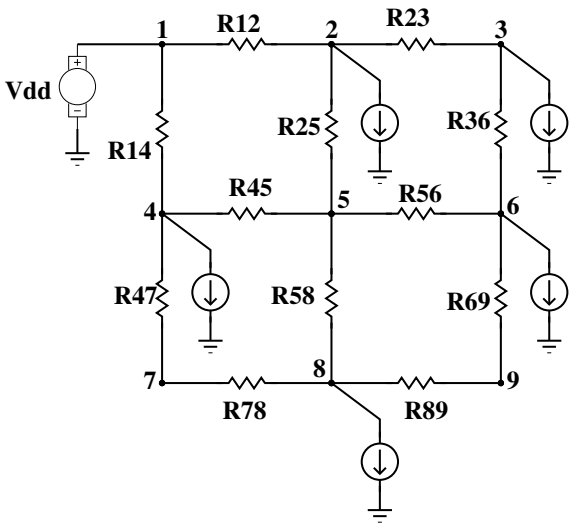


Figure 6.7: A PG network with 9 nodes and 12 edges

Proof. Here, N= 9, M=12, and let us consider $q = 1$ since the worst case IR drop noise occurs across $R14$ resistor in the Fig. 6.7 which is obtained by PG analysis. By doing PG analysis, we have got voltages of all the nodes as follows, $V_1 = 1.8V$, $V_2 = 1.5750V$, $V_3 = 1.5656V$, $V_4 = 1.6750V$, $V_5 = 1.6094V$, $V_6 = 1.5813V$, $V_7 = 1.6406V$, $V_8 = 1.6063V$, $V_9 = 1.5983V$.

Now, using (6.21), we get the value of η as follows,

$$\eta = 100.q.\frac{\binom{9}{1}}{\binom{9}{2}}\frac{V_{R14}}{V_{R14} + V_{R45} + V_{R47}} \tag{6.22}$$

$$\eta = 100.1.\frac{\binom{9}{1}}{\binom{9}{2}}\frac{125mV}{125mV + 65mV + 34mV} \tag{6.23}$$

$$\eta = 13.95 \tag{6.24}$$

Therefore, the number of mortal interconnects are,

$$\beta = \eta\%ofM \tag{6.25}$$

$$\beta = 13.95\%of12 \tag{6.26}$$

$$\beta = 1.674 \approx 2 \tag{6.27}$$

From this example we come to know that the PG network of Fig. 6.7 becomes dysfunctional when the two interconnects of the network becomes mortal. This analytical expression of η is used to find the MTTF of the PG network.

□

6.4.3 MTTF Prediction

The test set is generated by perturbing the same power grid benchmarks which are used for training data generation for validating the EM-aging ML model. Perturbation of the PG network is done for $\gamma_1\% = \gamma_2\% = 10\%$ as mentioned in the previous section for all the three cases. For the perturbed PG network, a similar procedure as mentioned in the Algorithm 6.1 (except for the output feature MTTF) is used to generate the test data set. Our trained model based on the Neural Network is tested using these test sets to predict MTTF of the PG network as mentioned in Algorithm 6.2. For the failure criterion of the PG network using our proposed model, we have selected MTTF of the first $\eta\%$ mortal interconnect as the MTTF of the PG network as mentioned in Theorem 6.1 and Theorem 6.2. To show the accuracy of the proposed model with the variation in the test set size, we compare the prediction accuracy with different test sets by varying the perturbation size ($\gamma_1\% = \gamma_2\%$) by 10%, 25%, 40%, 55%, 70%, 85% and 100% respectively for all the three cases (refer Fig. 6.10).

6.4.4 Identification of EM-affected Metal Segment of PG Network

Once the EM-aging prediction is completed, then it is necessary to detect the potential EM-affected metal segments of the PG network in order to obtain the reliable design of the PG network. For the identification of the potentially degraded metal segment, all the metal segments are numbered chronologically and labeled as mortal or non-mortal in the dataset based on the MTTF of the PG Network as determined by our proposed method mentioned in the previous section. We can also obtain the mortal interconnects in this stage of the simulation flow. However, if in the incremental design phase of the PG network, we want to create a model for identification of the mortal interconnects then we need to label the dataset and further create another machine learning-based classification model.Therefore, in view of this, a machine learning-based classification problem is formulated for classifying the degraded metal segments of PG Network. To accomplish this, we have used logistic regression which is a binary classification technique. We have used theses labeled dataset in order to train the classification model. The potential EM-affected metal segments can be identified using this binary classification technique from the trained model. The efficiency of the classification can be obtained using metrics such as precision, recall, and F1-score. These classification metrics are defined in the Table 6.2. For detecting potential EM-affected mortal PG interconnects, true positive means mortal interconnect is predicted as mortal, and true negative means mortal non-mortal interconnect is

predicted as non-mortal. Similarly, if mortal interconnect is predicted as non-mortal then it is called false positive, and if non-mortal interconnect is predicted as mortal then it is called as false negative. After the prediction using logistic regression-based classification, from the chronological numbers of the interconnects we obtain the predicted mortal interconnects. Subsequently, these interconnect designs are changed in order to make the power grid design EM resilient.

Table 6.2: Accuracy Metrics and its definition for logistic regression-based classification for detecting potentially EM-affected mortal PG interconnects.

Metrics	**Expression**
Accuracy	$\frac{\text{True Positive+True Negative}}{\text{True Positive+True Negative+False Positive+False Negative}}$
Precision	$\frac{\text{True Positive}}{\text{True Positive+False Positive}}$
Recall	$\frac{\text{True Positive}}{\text{True Positive+False Negative}}$
F1-Score	$2\times\frac{\text{Precision}*\text{Recall}}{\text{Precision+Recall}}$

6.5 Experimental Results

6.5.1 Simulation Setup

All the experiments are done using C++, and Python programming languages on a 3.2 GHz Linux based machine with 32 GB memory. For different machine learning models we have used scikit-learn [107] machine learning library. To test our proposed approach IBM power grid benchmark [3] and industry-based in-house power grid benchmarks are used. These PG benchmark statistics are mentioned in Table 6.4. The total simulation flow is shown in Fig. 6.8. Initially, from the power grid netlist feature extractions are done to extract J, L, T, IR drop, and MTTF of the PG interconnects. Power grid analysis using KLU Solver [12] is performed in order to obtain the currents and voltages of the power grid circuit and subsequently Black's equation is used. From which all the features are obtained and the dataset is prepared. Eventually, we train the neural network-based regression model for the model to learn the behavior of the dataset. For testing purposes, we have done the incremental power grid analysis which is a faster method. It facilitates the change in the current and voltages of the perturbed netlist in a faster way. Finally, the test dataset is tested on the neural network-based learned model and we get the predicted MTTFs of the PG interconnects. For each of the PG circuit, we have obtained r^2 score and MSE in order to verify the testing accuracy of the models. Once we get the predicted MTTFs of PG interconnects, we apply the new failure criterion of the PG network,

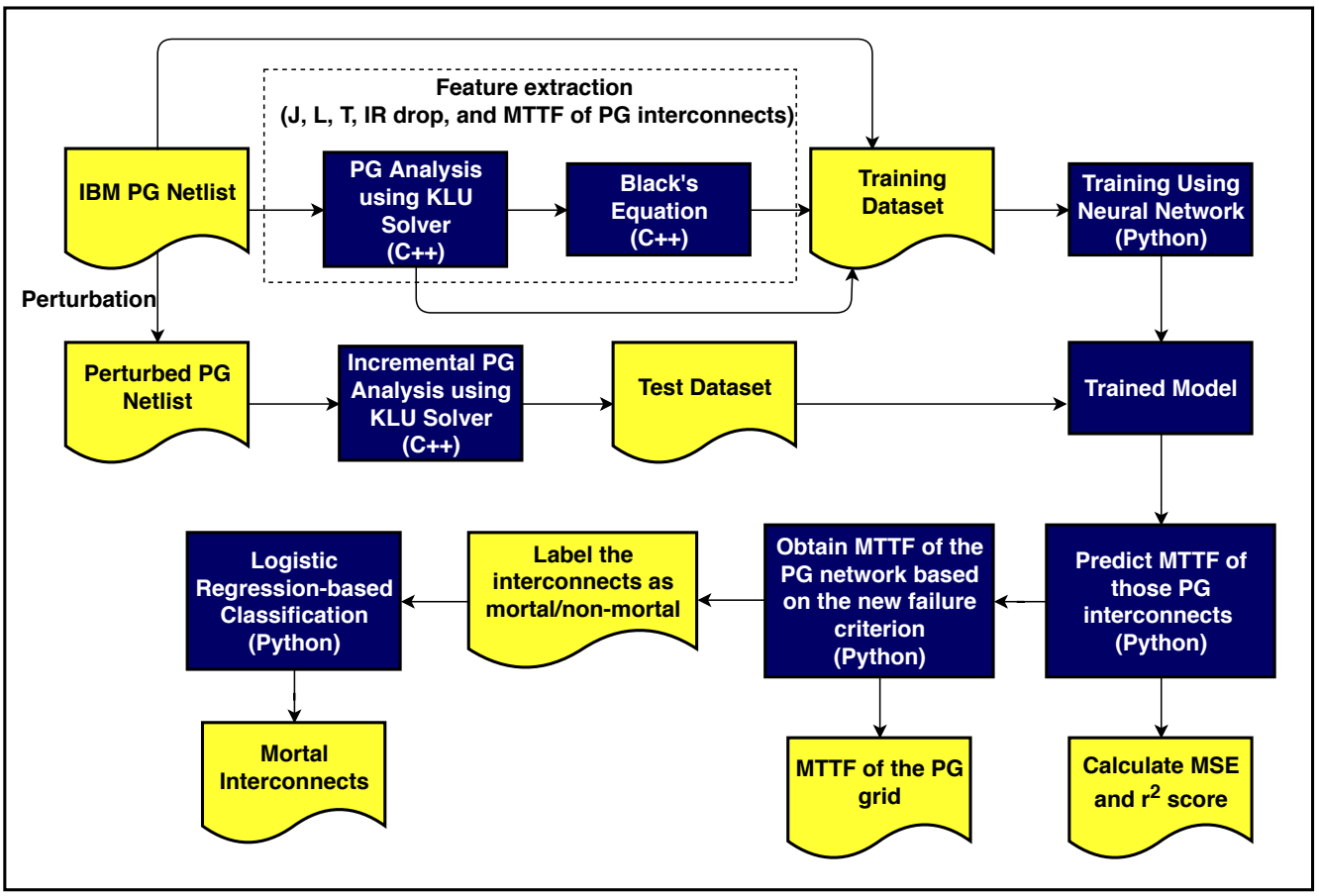


Figure 6.8: Flow of the simulation setup of the Machine Learning Flow

to find the MTTF of the PG network. Simultaneously, we have labeled the interconnects as mortal or non-mortal depending on its MTTFs and applied a logistic regression-based supervised learning technique to detect the potential mortal interconnects. All the results recorded in this section are averaged over 10 trails of each of the experiments.

Neural Network Architecture: The neural network architecture and the hyperparameters used in the experiments are described here. We have used ten hidden layers and Rectified Linear Unit (ReLU) activation function in our neural network architecture. Other hyperparameters of the neural network are listed in Table 6.3. The hyperparameters and the number of hidden layers are obtained after performing an exhaustive grid search using the `GridSearchCV` tool of the scikit-learn library. The notations used in Table 6.3 are standard notations used for regression models in scikit-learn library [107].

Table 6.3: Hyperparameters used in the Neural Network Architecture

Items	**Value**
#hidden layers	10
Activation	ReLU
Solver	Adam
α	0.001
batch size	Auto
learning rate	Constant
initial learning rate	0.01
$power\_t$	0.5
max_iter	1000
momentum	0.9
validation fraction	0.1
β_1	0.9
β_2	0.999
ϵ	1e-08

Our power grid analysis method using KLU Solver is faster than the industry standard *Synopsys HSPICE circuit simulator* [90]. The speedup of KLU solver over HSPICE is listed in Table 6.5. The reason behind the better speedup is KLU efficiently solves the linear system of equations resulting from the modified network analysis. KLU solver is used for circuit simulation in the feature selection phase. We have also used this KLU solver for incremental PG analysis in the test data generation phase.

Table 6.4: IBM PG benchmark [3] and industry-based PG benchmarks statistics

PG Circuits	**#n**	**#r**	**#v**	**#i**
*PG*1	10001	19800	200	9801
*ibmpg*1	30638	30027	14308	10774
*ibmpg*2	127238	208325	330	37926
*ibmpg*3	851584	1401572	955	201054
*ibmpg*4	953583	1560645	962	276976
*ibmpg*5	1079310	1076848	539087	540800
*ibmpg*6	1670494	1649002	836239	761484
*ibmpgnew*1	1461036	2352355	955	357930
*ibmpgnew*2	1461039	1422830	930216	357930
*PG*2	1000001	1998000	2000	998001
*PG*3	4000001	7996000	4000	3996001
*PG*4	9000001	17994000	6000	8994001

#n : total number of vertices (nodes) of PG network, **#r** : total number of resistors (edges) of PG network, **#v** : total number of supply voltage (V_{dd} and GND supply source) of PG network, **#i** : total number of current workloads connected to PG network.

Table 6.5: Comparison of circuit analysis time for HSpice and KLU Solver

	Time (sec)		**Speedup**
PG circuits	**HSPICE**	**KLU Solver**	$\frac{\textbf{Time}_{HSPICE}}{\textbf{Time}_{KLU}}$
*PG*1	0.90	0.15	6.00×
*ibmpg*1	2.85	0.56	5.08×
*ibmpg*2	19.46	2.61	7.45×
*ibmpg*3	29.60	5.07	5.73×
*ibmpg*4	54.4	5.83	9.33×
*ibmpg*5	73.80	7.74	9.53×
*ibmpg*6	96.5	10.10	9.55×
*ibmpgnew*1	103.58	14.50	7.14×
*ibmpgnew*2	47.60	8.86	5.37×
*PG*2	67.14	12.85	5.22×
*PG*3	247.36	49.80	4.96×
*PG*4	587.60	120.37	4.88×

Table 6.6: Training and Testing time for power grid benchmarks using Neural Network

PG circuits	t_1 **(s)**	t_2 **(s)**	t_3 **(s)**	t_4 **(s)**	t_{ML} **(s)**	t_{ML} **(min)**	t_{ML} **(hr)**
$PG1$	0.30	0.001	0.069	0.001	0.371	0.006	0.0001
$ibmpg1$	1.10	0.001	0.697	0.002	1.800	0.021	0.0003
$ibmpg2$	5.20	0.01	1.088	0.006	6.304	0.105	0.002
$ibmpg3$	10.25	0.02	21.378	0.179	31.827	0.530	0.009
$ibmpg4$	11.33	0.02	12.476	0.165	23.991	0.390	0.007
$ibmpg5$	15.62	0.03	8.810	0.096	24.196	0.400	0.006
$ibmpg6$	21.52	0.03	13.174	0.147	34.871	0.580	0.009
$ibmpgnew1$	28.46	0.04	18.920	0.220	47.640	0.790	0.013
$ibmpgnew2$	16.78	0.03	13.079	0.129	30.018	0.500	0.008
$PG2$	25.47	0.03	10.611	0.132	36.243	0.600	0.010
$PG3$	100.45	0.05	50.663	0.676	151.839	2.530	0.042
$PG4$	245.89	0.19	117.216	1.557	364.853	6.080	0.101

* t_1 = training data generation time including power grid analysis time using KLU solver and Black's model implementation time in seconds, t_2 = test data generation time including incremental power grid analysis time in seconds, t_3 = training time in seconds, t_4 = testing time including prediction time in seconds, $t_{ML} = t_1 + t_2 + t_3 + t_4$, t_{ML} (s) = total time required summing all the times in secs, t_{ML} (min) = t_{ML} converted in minutes, t_{ML} (hr) = t_{ML} converted in hour.

6.5.2 Total Time Required for the Machine Learning Model

The training data generation time including the circuit simulation time for the different IBM power grid benchmarks is given in Table 6.6. Our main aim is to train the data of a certain PG network and to predict the MTTF of the PG network for any incremental changes in the PG design. For the training data generation, we used the KLU solver to solve the $GV = I$ matrix. However, for the test data generating, we have used the incremental analysis using KLU solver for the incremental PG design, to speed up the process as mentioned in Section 6.3.4. To show that we have done the perturbations in the PG network design and predict the MTTF by our trained ML-model.

6.5.3 Results of Expected number of Mortal interconnects of PG Network

The expected number of mortal interconnects (η of (6.21)) of the PG network until the PG network is considered dysfunctional is calculated using for all the PG benchmarks using the Theorem 6.1. We consider three instances of worst-case IR drop while evaluating the value of η by varying the value of q. It is observed that as the value of q increases from 1 to 100, the number of mortal interconnects until the failure of the PG network increases. The results of η for different PG benchmarks are shown in Fig. 6.9. This result is shown in order to observe the practical values of η for the PG benchmarks.

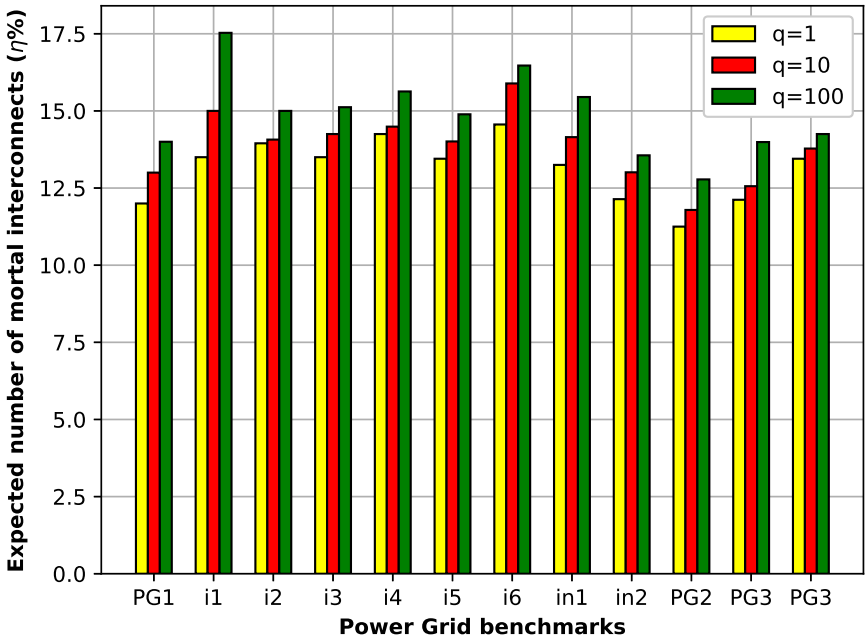


Figure 6.9: Expected number of mortal interconnects (η% of total PG interconnects) for different PG benchmarks until the PG network is considered dysfunctional. Here $i1$, $i2$, $i3$, $i4$, $i5$, $i6$ $in1$, and $in2$ denote $ibmpg1$, $ibmpg2$, $ibmpg3$, $ibmpg4$, $ibmpg5$, $ibmpg6$, $ibmpgnew1$, and $ibmpgnew2$ benchmark respectively.

6.5.4 Results of MTTF Prediction and CPU Runtime

This section demonstrates the improvement in the predicted value of MTTF and CPU runtime of the EM aging prediction model. The results of MTTF and CPU runtime for all the power grid benchmarks are listed in Table 6.7 and Table 6.8 respectively. We have compared our results with the works of Chatterjee et al. [5,6], Huang et al. [4] and Najm et al. [7], which are also listed in Table 6.7 and Table 6.8. We have got a significant improvement in the speedup for our machine learning based EM-aging prediction approach over the work of [5,6], [4], and [7] as listed in the Table 6.8. The reason for this significant speedup of our ML-based approach is that the time taken for training dataset generation, to train the model, and predicting the MTTFs for the test dataset is much lesser than those physics-based state-of-the-art models. The physics-based models solve the PDE (corresponding to the interconnects) and solve the equations for the whole power grid network to obtain the MTTF of the power grid network. Generally, the solutions are obtained using iterative PDE solver or some approximation techniques. As a result, the execution time takes a considerable amount of time to converge. In [4], the authors adopted a similar methodology, as shown in Fig. 6.1. The model of [4] is

Table 6.7: Comparison of MTTF for our proposed ML-based approach with works of [4–7] for IBM power grid benchmarks.

	MTTF (μ) (years)				
Methods	**TCAD2016 [4]** (μ_H)	**ICCAD2017 [5]** (μ_{Ch})	**TCAD2018** [6] (μ_C)	**IRPS2019** [7] (μ_N)	**Proposed** (μ_{ML})
PG Circuits					
*PG*1	14.01	6.10	8.51	6.5	13.25
*ibmpg*1	12.55	6.50	10.91	7.0	12.10
*ibmpg*2	18.75	6.78	10.11	12.1	12.55
*ibmpg*3	31.96	6.66	9.95	6.7	12.25
*ibmpg*4	33.39	9.83	11.95	16.7	17.48
*ibmpg*5	25.16	6.54	6.63	6.3	10.33
*ibmpg*6	19.87	9.53	11.96	11.2	12.41
*ibmpgnew*1	25.96	13.24	11.64	13.2	14.56
*ibmpgnew*2	21.80	5.72	6.72	7.3	13.24
*PG*2	17.85	8.32	9.32	10.3	11.21
*PG*3	-	-	-	7.2	10.51
*PG*4	-	-	-	6.8	8.47

Table 6.8: Comparison of CPU Runtime for our proposed ML-based approach with works of [4–7] for IBM power grid benchmarks.

	CPU Runtime (t) (Hours)					**Speedup**			
Methods	**TCAD2016** [4] (t_H)	**ICCAD2017** [5] (t_{Ch})	**TCAD2018** [6] (t_C)	**IRPS2019** [7] (t_N)	**Proposed** (t_{ML})	$\frac{t_H}{t_{ML}}$	$\frac{t_{Ch}}{t_{ML}}$	$\frac{t_C}{t_{ML}}$	$\frac{t_N}{t_{ML}}$
PG Circuits									
*PG*1	0.02	0.02	0.001	0.000166	0.0001	200×	200×	10×	1.66×
*ibmpg*1	0.05	0.03	0.003	0.01000	0.0003	166.66×	100×	10×	33.33×
*ibmpg*2	0.11	0.31	0.04	0.02000	0.002	55×	155×	20×	10×
*ibmpg*3	5.83	4.27	0.41	0.07000	0.009	647.77×	610×	45.55×	7.77×
*ibmpg*4	14.71	6.81	2.31	0.11000	0.007	2101.42×	972.85×	330×	15.71×
*ibmpg*5	0.69	0.25	0.06	0.03000	0.006	115×	41.66×	10×	5×
*ibmpg*6	1.75	2.07	0.79	0.23330	0.009	194.44×	230×	87.77×	25.92×
*ibmpgnew*1	16.78	0.42	1.24	0.08000	0.013	1290.76×	32.06×	95.38×	6.15×
*ibmpgnew*2	15.32	2.60	0.43	0.06000	0.008	1915×	325×	53.75×	7.50×
*PG*2	10.94	1.12	1.06	0.10166	0.010	1094×	112×	106×	10.06×
*PG*3	-	-	-	0.13666	0.04200	-	-	-	3.25×
*PG*4	-	-	-	0.25666	0.10100	-	-	-	2.54×
Avg. Speedup						778×	277.85×	76.84×	10.74×

one of the most accurate EM evaluation models, as the authors analytically solve the PDEs. However, this approach takes a huge time. For large PG circuits, this model doesn't converge due to limitations of the system memory. In [5], authors have extended the existing physics-based models for EM evaluation and represented as linear time-invariant (LTI) systems. This LTI system is solved iteratively to get the MTTF. This method also takes a considerable amount of time and system memory. In [6], authors further extended their LTI system-based EM assessment approach by incorporating macro-modeling-based filtering and predictor approach. However, in our ML-based EM-aging model, we train the model with historical data, and testing is done using a test dataset, which does not take much time. As a result, we get significant speedups compared to [4–6] (Please refer to Table 6.8). The results of [4–6] are reproduced using the similar EM model parameters as reported in [4–6]. The MTTF prediction results using our machine learning approach is also better than that of [5, 6] and comparable to the accurate physics-based model of [4]. Our predicted MTTF values are better than [5,6] and comparable to [4], because of our new EM-failure criterion which we discussed in Section 6.4.

Due to the large size of the PG3 (∼4M nodes and ∼7.9M interconnects) and PG4 (∼9M nodes and ∼17.9M interconnects) circuits, the models of [4–6] need huge system memory and don't converge on our system with 32 GB memory and 3.2 GHz processor. All the three models [4–6] generate interconnect trees from the PG netlist. Subsequently, the solution is obtained by solving the PDEs corresponding to interconnect trees iteratively. Generating the solution of PDEs for large PG circuits makes the MTTF computation expensive in terms of execution time and system memory. As a result, our implementation of the physics-based models [4–6] does not converge on our system for the PG3 and PG4 circuits. This also shows the scalability of our proposed machine learning model for large scale PG circuits.

Recent work by Najm et al. [7] shows a significant speedup over the work of [4–6] for MTTF prediction of PG network. They have used Monte-Carlo random sampling for predicting the MTTF of a large scale PG network. We also obtained the results using EM models of [7] and listed it in Table 6.7 and Table 6.8. From the results, it can be observed that our proposed approach also outperforms the results of [7] in terms of both speedup and EM lifetime (our EM lifetime is much closer to the EM lifetime of [4]). Therefore, it can be proved that the CPU runtime of EM-aging prediction using our machine learning method is faster than the work of [7]. Even the value of MTTF predicted by our EM-aging model with the new failure criterion gives an improvement in MTTF than that of [7].

Therefore, deploying this machine learning approach decreases the EM sign off time significantly with an estimated error (MSE) of less than 3% for a perturbation size of 10% (see Figure 6.10). More about MSE and perturbation size are given in the following results.

It is worth mentioning that the work [4], which was published in 2016, presented an accurate method. However, the method of [4] is time and computing resource-consuming and not feasible to adapt in real-world full-chip EM lifetime prediction of large-scale power grid circuit. Please note that the work [5–7], which was subsequently published in 2017, 2018, 2019 were not accurate and have significant speedup compared to [4]. Our proposed ML-based approach is faster than [5–7], and EM lifetime results are much closer to [4] than that of [5–7]. With our proposed approach, it will be easier for the designers to get an approximate estimation of the EM lifetime in very less time, which makes the total design cycle faster.

6.5.5 Results of Regression Model Prediction Accuracy

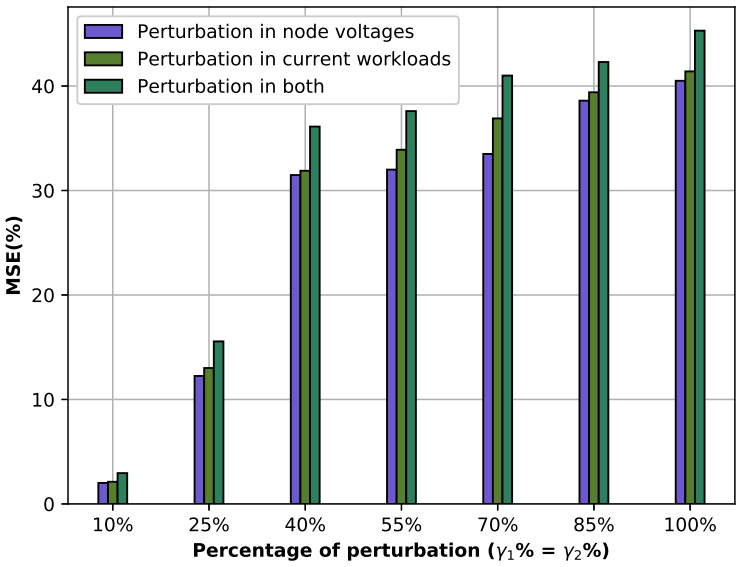
This experiment is done to show the comparison of accuracy of the proposed EM-aging prediction model using Neural Network (NN) over the other well-known regression techniques which includes Bayesian linear regression [103], Random Forest regression [108], Ridge linear regression [109], SVM regression [110], Gaussian Process Regression (GPR) [111] are shown in this section. The metrics used to compare the accuracy are the r^2 score and the mean-square-error (MSE) for all the test sets generated from the IBM power grid benchmarks [3]. A higher value r^2 score (≤ 1) denotes better compactness of the data to the regression model. The r^2 score and MSE value of the test set for all the power grid benchmarks are listed in the Table 6.9, and Table 6.10 respectively. Simultaneously, the results are also compared with all other standard regression techniques with their best possible settings for prediction. The results show that for all the seven IBM power grid benchmarks, the Neural Network model outperformed other models (see Neural Network column in Table 6.9 and Table 6.10). Although, the GPR regression technique gives a better value of r^2 score and MSE for two benchmark circuits. However, GPR needs huge runtime memory requirement and hence it is not able to give any feasible result for the larger PG benchmarks (*ibmpg*2 onwards). From this experiment, we want to prove that our EM-aging model for PG network using the Neural Network works best compared to the other standard regression models in terms of r^2 score and MSE value.

Table 6.9: MSE of the EM aging models tested by different regression techniques For IBM power grid benchmarks.

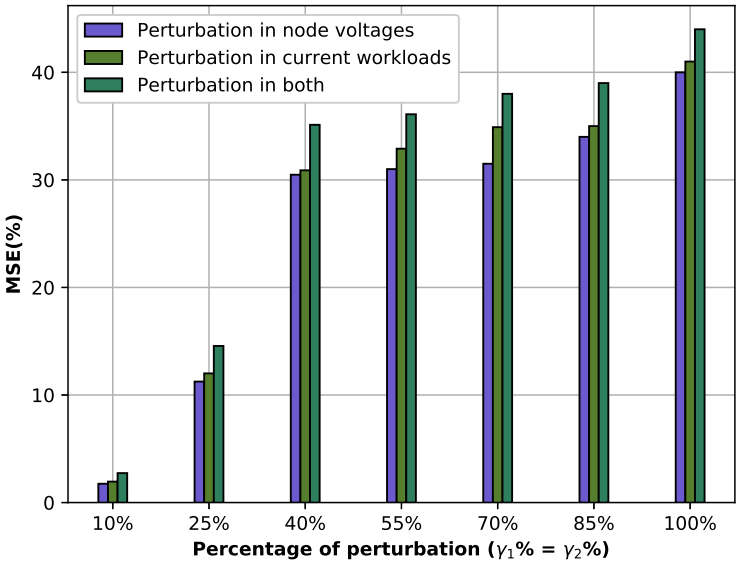
PG Circuits	#interconnects	Mean Square Error					
		Bayesian	**Random Forest**	**Ridge**	**SVM**	**Gaussian**	**Neural Network**
*PG*1	10001	3.69%	5.97%	3.71%	8.35%	2.42%	2.99%
*ibmpg*1	30027	3.66%	5.93%	3.61%	8.23%	2.32%	2.95%
*ibmpg*2	208325	4.25%	6.24%	3.92%	10.45%	-	3.11%
*ibmpg*3	1401572	4.52%	7.56%	4.21%	12.12%	-	2.74%
*ibmpg*4	1560645	3.95%	6.67%	3.75%	9.25%	-	2.89%
*ibmpg*5	1076848	4.14%	5.92%	3.83%	9.72%	-	3.31%
*ibmpg*6	1649002	3.93%	6.54%	3.89%	10.37%	-	3.20%
*ibmpgnew*1	2352355	4.23%	6.89%	3.75%	8.52%	-	3.15%
*ibmpgnew*2	1422830	4.15%	6.64%	3.78%	9.25%	-	3.01%
*PG*2	1000001	3.96%	6.52%	3.69%	8.24%	-	2.97%
*PG*3	4000001	3.75%	5.98%	3.36%	8.11%	-	2.52%
*PG*4	9000001	3.25%	5.91%	3.24%	8.02%	-	2.23%

Table 6.10: r^2 score of the EM aging models tested by different regression techniques For IBM power grid benchmarks.

PG Circuits	#interconnects	r^2 score					
		Bayesian	**Random Forest**	**Ridge**	**SVM**	**Gaussian**	**Neural Network**
*PG*1	10001	0.932	0.941	0.963	0.825	0.996	0.994
*ibmpg*1	30027	0.971	0.952	0.971	0.855	0.997	0.995
*ibmpg*2	208325	0.937	0.926	0.935	0.832	-	0.986
*ibmpg*3	1401572	0.920	0.914	0.917	0.815	-	0.975
*ibmpg*4	1560645	0.921	0.935	0.921	0.821	-	0.977
*ibmpg*5	1076848	0.926	0.942	0.925	0.829	-	0.983
*ibmpg*6	1649002	0.929	0.925	0.926	0.828	-	0.981
*ibmpgnew*1	2352355	0.921	0.911	0.922	0.823	-	0.978
*ibmpgnew*2	1422830	0.916	0.922	0.914	0.814	-	0.975
*PG*2	1000001	0.922	0.932	0.951	0.855	-	0.995
*PG*3	4000001	0.936	0.941	0.965	0.857	-	0.997
*PG*4	9000001	0.941	0.945	0.971	0.856	-	0.998



(a)



(b)

Figure 6.10: Comparison of prediction accuracy on test set in MSE with variations in perturbations size for (a) *ibmpg*2 (b) *ibmpg*4 benchmark circuit.

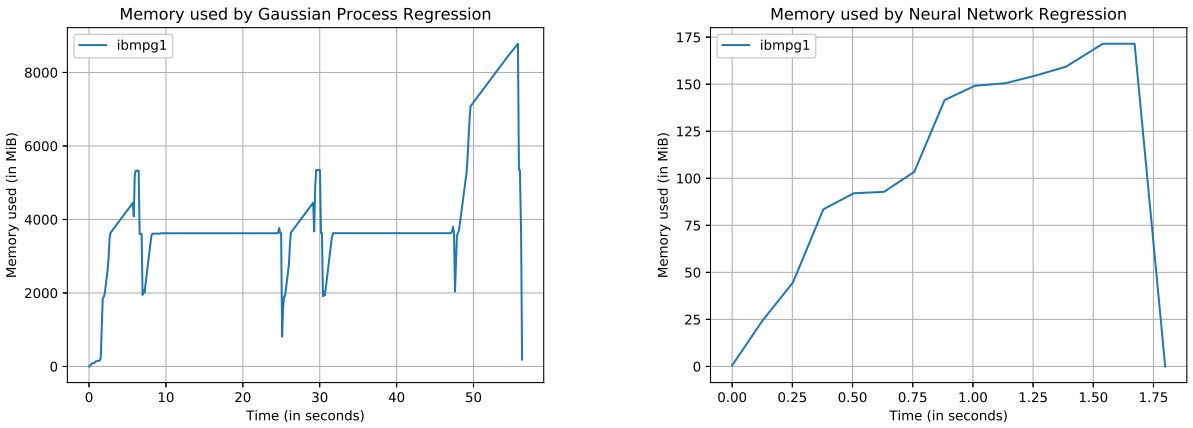
6.5.6 Comparison of MSE with Variations in Perturbation Size

This section demonstrates the test accuracy of the EM-prediction model in the incremental design of the PG network by varying the perturbation size of the test set. For the incremental design of the power grid network, usually, the changes or the perturbations done in designs in each iterative step is much less than 10% [19]. Therefore, for all the experiments throughout the manuscript, we have kept the perturbation size constant at 10%. However, in order to see the test accuracy for larger perturbation, in this section, we have used different perturbation size from 10%, 25%, 40%, 55%, 70%, 85%, and 100% and verify the MSE for different PG benchmarks data. Results of the comparison of prediction accuracy on the test set in MSE with variations in perturbations size for *ibmpg*2 and *ibmpg*4 circuit is shown in Fig. 6.10. It can be understood from Fig. 6.10 that the MSE of our EM-aging prediction model increases as the size of the perturbation size increases. From this experiment, we want to reiterate the fact that our EM-aging prediction model is best for those incremental designs of the PG network where the perturbations are the least (much less than 10%). We also want to show that as the size of the perturbation increases the EM-aging prediction accuracy decreases. Therefore, our proposed EM-aging prediction model is only applicable to those incremental designs where little perturbations (around 10%) are considered.

6.5.7 Comparison of Peak Memory and CPU runtime of ML models

In this section, we have shown the memory used by our EM aging model using Neural Network model. We have also done the comparative study of the memory used by EM aging model with different regression techniques and listed the peak memory result for each of the benchmark circuits in Table 6.11. Although the Gaussian Process Regression (GPR) technique is believed to be the best regression technique due to least MSE value and a r^2 score closer to 1 compared to the other regression technique, as shown in previous section. It is observed that the GPR takes huge memory compared to the other regression models. Due to the huge memory requirement, it has failed to give any result for large benchmark circuits in our machine with 32 GB memory. For *ibmpg*1 circuit memory used during runtime by the EM aging model with GPR and Neural Network technique is shown in Fig. 6.11. We can observe from the figure that the peak memory used by GPR is more than 8000 Mebibyte (MiB), whereas peak memory used by Neural Network is 174 MiB. The reason behind huge memory taken by the GPR technique is that it stores data in matrices form and do matrix computation to predict

the output feature.



(a) (b)

Figure 6.11: Memory used by EM aging model during MTTF prediction of *ibmpg*1 benchmark circuit using (a) GPR model and (b) Neural Network model. It is to be noted 1 Gigabyte (GB) = 953.674 Mebibyte (MiB)

Table 6.11: Peak memory used by EM aging prediction models using different regression techniques For IBM power grid benchmarks.

PG Circuits	**#interconnects**	**Peak memory (in MiB)**					
		Bayesian	**Random Forest**	**Ridge**	**SVM**	**Gaussian**	**Neural Network**
*PG*1	10001	172.75	173.59	172.95	172.21	1146.00	173.09
*ibmpg*1	30027	174.60	174.41	174.80	173.89	8786.51	174.96
*ibmpg*2	208325	205.52	203.90	205.48	280.69	-	205.61
*ibmpg*3	1401572	533.14	506.62	501.26	845.75	-	523.46
*ibmpg*4	1560645	575.96	541.58	540.44	898.79	-	565.51
*ibmpg*5	1076848	444.94	442.87	445.15	733.07	-	487.08
*ibmpg*6	1649002	601.45	557.51	563.92	932.08	-	590.18
*ibmpgnew*1	2352355	792.03	729.89	738.39	1178.69	-	774.69
*ibmpgnew*2	1422830	535.95	502.84	503.63	848.76	-	520.94
*PG*2	1000001	673.81	657.38	628.37	1037.5	-	667.37
*PG*3	4000001	2208.02	2092.046	1964.62	3090.55	-	2208.94
*PG*4	9000001	3594.94	4072.86	3183.58	5333.87	-	3595.95

Similarly, it can be seen that the time taken by the GPR technique is also very high in as shown in Table 6.12 for the *PG*1, and *ibmpg*1 benchmark circuits. We can also observe that the Bayesian and Ridge regression technique take less time than the Neural Network. However, the Bayesian and Ridge regression techniques don't give accurate results as shown in the Table 6.9 with higher value of MSE. These two techniques don't fit properly for this EM-aging prediction model compared to the Neural Network which is also known from the r^2 score as listed in Table 6.9.

Table 6.12: CPU Runtime of the EM aging models tested using different regression techniques For IBM power grid benchmarks.

PG Circuits	#interconnects	CPU Runtime (in Seconds)					
		Bayesian	Random Forest	Ridge	SVM	Gaussian	Neural Network
*PG*1	10001	0.307	0.814	0.308	0.322	4.644	0.371
*ibmpg*1	30027	1.106	1.852	1.106	1.165	54.876	1.800
*ibmpg*2	208325	5.240	7.602	5.222	6.249	-	6.304
*ibmpg*3	1401572	10.37	30.052	10.336	32.659	-	31.827
*ibmpg*4	1560645	11.492	35.098	11.418	50.125	-	23.991
*ibmpg*5	1076848	15.739	31.160	15.715	28.565	-	24.196
*ibmpg*6	1649002	21.665	46.253	21.628	61.127	-	34.871
*ibmpgnew*1	2352355	28.711	65.250	28.607	85.260	-	47.64
*ibmpgnew*2	1422830	16.910	35.818	16.879	51.593	-	30.018
*PG*2	1000001	25.623	50.766	25.581	74.736	-	36.243
*PG*3	4000001	101.001	201.612	100.788	357.095	-	151.839
*PG*4	9000001	247.149	478.847	246.717	926.052	-	364.853

6.5.8 Results of EM-affected Metal Segment Identification

The results of logistic regression-based identification of the EM-affected metals segments are listed in Table 6.13. Different metrics are mentioned to obtain the prediction quality of the logistic regression-based classification model. From the accuracy metrics, we know about the total accuracy of the model. As the size of the dataset increases the classification accuracy also increase. Therefore, for large power grid benchmark circuits the classification accuracy is higher. The Precision metric is a good measure to determine the number of false positives as mentioned earlier. The Recall metric states the number of actual positives classified by the model. In order to obtain a balance between precision and recall, F1-score is obtained. For the results we have got for classification, almost for all the power grid benchmarks, the metrics obtained are good enough for detecting the mortal interconnects.

Table 6.13: Results for accuracy of the logistic regression-based EM-affected metal segment identification.

PG Circuits	Classification Accuracy	Precision	Recall	F1-score
*PG*1	0.9701	0.9844	0.9798	0.9821
*ibmpg*1	0.9715	0.9872	0.9781	0.9826
*ibmpg*2	0.9750	0.9868	0.9857	0.9863
*ibmpg*3	0.9800	0.9899	0.9877	0.9888
*ibmpg*4	0.9810	0.9910	0.9879	0.9893
*ibmpg*5	0.9805	0.9895	0.9873	0.9882
*ibmpg*6	0.9865	0.9910	0.9938	0.9924
*ibmpgnew*1	0.9965	0.9972	0.9988	0.9980
*ibmpgnew*2	0.9886	0.9911	0.9939	0.9925
*PG*2	0.9975	0.9983	0.9988	0.9986
*PG*3	0.9983	0.9994	0.9988	0.9991
*PG*4	0.9991	0.9994	0.9994	0.9994

6.6 Conclusion

This chapter presents an approach to predict the electromigration (EM)-based aging of the on-chip power grid (PG) network using a machine learning method. Neural Network regression-based machine learning technique is used to predict the mean-time-to-failure (MTTF) during the incremental design of the PG network. The training set is generated using the parameters of the PG network, and appropriate features are selected for the proposed machine learning approach by evaluating r^2 score. To speed up the training data generation phase, KLU solver is used for power grid analysis in order to extract the features. For generating the test dataset, a perturbation in the PG network is done, and incremental power grid analysis using KLU solver is used to speed up the process. The trained model is used on the test set for MTTF prediction of different power grid benchmarks. A new failure criterion is proposed in order to improve the MTTF of the proposed model. Results on different power grid benchmark circuits show that the proposed machine learning model exhibits a significant speedup than all of the state-of-the-art EM-based MTTF prediction models. The MTTF predicted by our proposed model is also better than some models and comparable to the most accurate model. Further, we have also proposed a logistic regression-based classification model in order to detect potentially degraded PG interconnects. We also demonstrated different performance and accuracy metrics of the Neural Network model with other standard regression methods in terms of r^2 score, mean-square-error (MSE), CPU runtime, and peak memory used. Neural Network is found to be the best among all the models.

From our work and experimental results, we can recommend the following key points,

- Machine learning-based approach can be applicable for EM aging prediction in incremental PG design.
- Neural network-based supervised machine learning model is the best among all well-known machine learning technique, for the EM aging prediction in incremental PG design in terms of r^2 score, MSE, and peak memory consumption metrics.
- A significant speedup over the state-of-the-arts works can be achieved using the machine learning approach.
- The MTTF value obtained using our machine learning model is most close to the accurate physics-based model reported in the literature, and compared to the other state-of-the-art models.

- Speeding up the MTTF prediction process helps in overall design sign-off time.
- For larger perturbations in the test set, the machine learning technique incurs a significant MSE. As a result, our proposed machine learning model is only applicable for EM aging prediction in incremental PG design.

In the next chapter, all contributions of the thesis are summarized and future research direction is described.

7

Conclusions and Future Works

Contents

7.1 Introduction 130
7.2 Summary of the Contributions 130
7.3 Limitation 132
7.4 Future Works 132

7.1 Introduction

The work of this thesis is motivated towards improving the on-chip power grid design methodology with Artificial Intelligence and Machine Learning techniques as viable options. Towards this, we work on major two design challenges of the on-chip power grid design phase. These two challenges are IR drop and Electromigration issues. Both of these issues increase failure probability of the power grid network as well as the chip. Existing works mostly solve the IR drop analysis with linear algebraic methods which is a time-consuming process for the large power grid networks. Also, it is necessary to optimize the power grid design considering various critical design objectives. Existing works of literature do not address these multiobjective optimization issues, instead, the work of literature only considers area minimization as the power grid optimization solution. The use of simple linear programming techniques for optimizing the power grid is also not a good option for large power grid networks. Further, it is necessary to obtain the electromigration-aware aging prediction of the power grid networks, during the design phase itself. Existing physics-based approaches take a large amount of time for design sign-off. Therefore, for all the problems of power grid design, a fast solution is required. We have discovered that the AI/ML techniques help in fast sign-off of the power grid design problems. This chapter concludes all the proposed contributions of this thesis along with the future directions for research.

7.2 Summary of the Contributions

- **Power Grid Analysis using Probabilistic Approach:** This contribution presents a power grid analyzer based on Lévy flight principle. In this work, Lévy flight approach is used to for traversing the power grid network, which helps in obtaining a fast solution of the power grid network. We also remove the self loops in our proposed approach, which helps in fast convergence in power grid network solutions. Our proposed approach is validated using large-scale power grid benchmarks. Results show significant speedup over the Random walk and Gauss Seidel approach.
- **Design Space Exploration of Power Grid using Heuristic Approach:** In this contribution, a multi-objective framework for minimization of the IR drop-metal routing area of the VLSI PGN is proposed using evolutionary computation technique. Initially, the power grid design process and all reliability issues during the design phase are described. Subsequently, the objective function considering the IR drop and metal routing area is formulated with the consideration of reliability and yield-based design constraints. NSGA-II based multi-objective evolutionary

algorithms have been employed to minimize the two objectives of the problem simultaneously and to obtain an optimum point of trade-off. Experimental results on standard IBM power grid benchmarks show that our proposed framework is able to obtain an optimum point for the IR drop and metal routing area.

- **Power Grid Design using Machine Learning:** In this contribution, we have proposed a deep learning-based framework to predict the initial power grid design. We predict the power grid interconnect width as part of the design process, which is time-consuming and tedious work. Subsequently, we also predict the worst-case IR drop in the power grid. A neural network-based multi-regression technique is used in our model for accomplishing the prediction tasks. Results on IBM power grid benchmarks show $\sim$6$\times$ speedup than the conventional power grid design approach.

- **Aging Prediction of Power Grid Design using Machine Learning:** This contribution presents an approach to predict the electromigration (EM)-based aging of the on-chip power grid (PG) network using a machine learning method. Neural Network regression-based machine learning technique is used to predict the mean-time-to-failure (MTTF) during the incremental design of the PG network. The training set is generated using the parameters of the PG network, and appropriate features are selected for the proposed machine learning approach by evaluating r^2 score. For generating the test dataset, a perturbation in the PG network is done. The trained model is used on the test set for MTTF prediction of different power grid benchmarks. A new failure criterion is proposed in order to improve the MTTF of the proposed model. Results on different power grid benchmark circuits show that the proposed machine learning model exhibits a significant speedup than all of the state-of-the-art EM-based MTTF prediction models.

The summary of the thesis is shown in Fig. 7.1. From the work of Chapter 5 and Chapter 6, it is also observed that machine learning approaches work well for incremental designs, where iteratively little perturbation is made in designs. In such iterative cases of design, machine learning can be utilized efficiently to predict instantly the changes that occur in the design, instead of performing all the simulations and design exploration all over again. Further, in order to fully utilize the benefit of machine learning in the design phase of on-chip power grid interconnect, more robust learning techniques need to be explored.

7.3 Limitation

The work proposed in Chapter 3 is applicable to regular large power grids. In order to adapt the proposed approach of Chapter 3 for practical cases of power grid circuits, it is necessary to have an analytical equivalent resistance model for practical non-uniform power grids.

The methods presented in Chapter 4 and Chapter 5 are designed with the assumption that it is a two-layer power grid. However, the work can be extended for a multi-layer power grid with proper formulation and calibration.

As mentioned before, Âăthe machine learning approaches are found to produce good results for incremental designs. Therefore, further work is required in order to design fully automated machine learning solutions for on-chip power grid design.

7.4 Future Works

The contributions of this thesis can be extended in several ways. Some of the possible future research directions are listed below:

- The machine learning-based proposed works in this thesis uses manual feature engineering. In future, automatic feature engineering can be employed to further automating the learning process of the power grid design.
- This thesis profoundly concentrated on formulating the problems as supervised learning problems. In the future, power grid design problems can be formulated as unsupervised learning problems and solved using the emerging learning approaches such as Variational Autoencoder, Generative Adversarial Network etc.
- Other objectives of the power grid design can further be solved using AI/ML approaches.
- Parallelization techniques can be explored for fast sign-off of the power grid analysis.
- Thermal Issues of PG Design can be explored.
- Extension of the works to PG Design of 3D IC.
- This thesis's proposed methods can also be extended to electrical grid design (used for delivering electricity from producers to consumers), with appropriate changes.

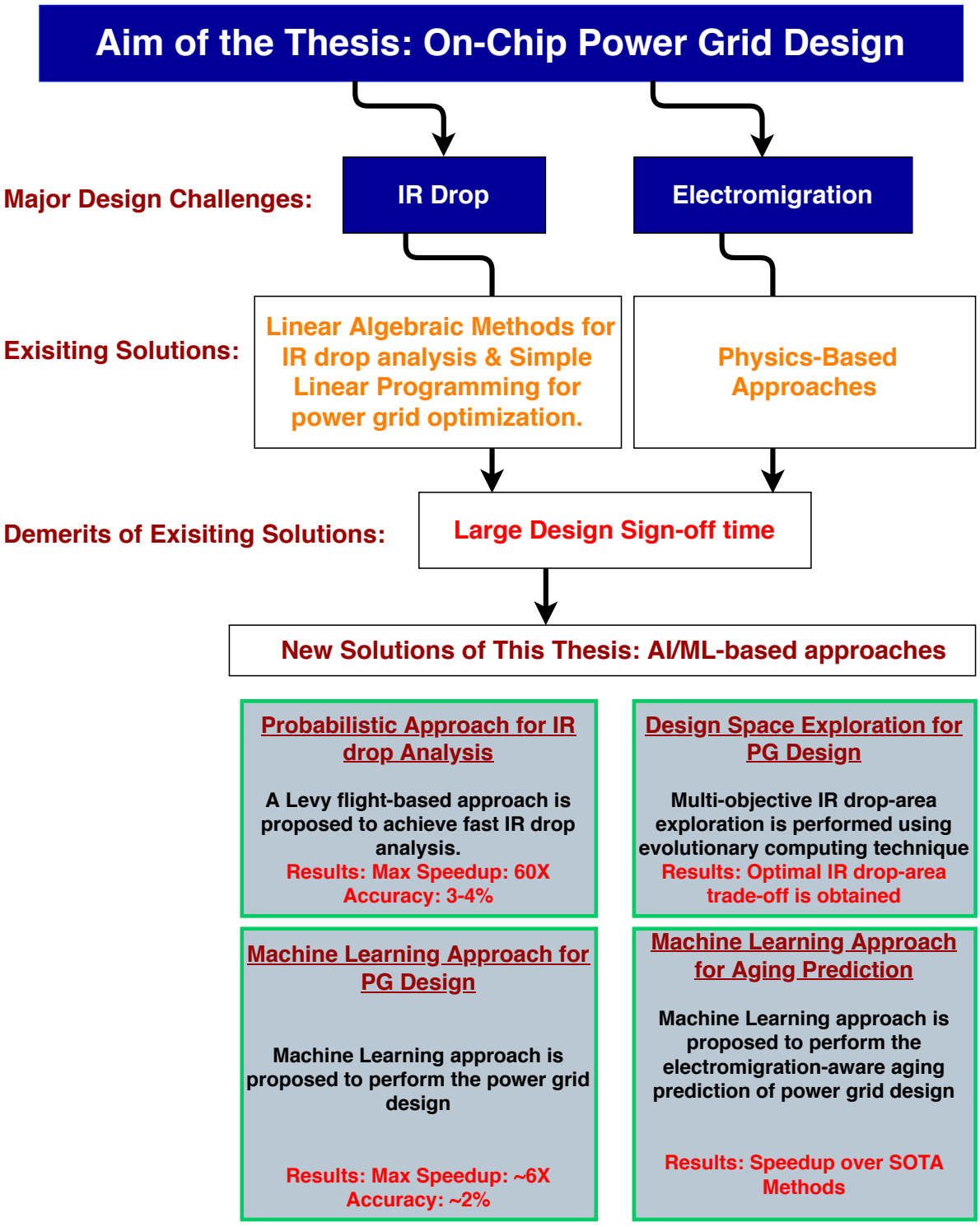


Figure 7.1: Summary of the Thesis

A

Power Grid Benchmarks

Contents

A.1 IBM Power Grid Benchmarks 137
A.2 SPICE Netlist Generation 137
A.3 Sample IBM Power Grid Benchmark Netlist 138
A.4 IBM Power Grid Benchmark Statistics 142
A.5 Large Power Grid Benchmarks 143

A.1 IBM Power Grid Benchmarks

The IBM power grid benchmarks used in these for experimental purpose is taken from [112]. These power grid benchmarks are available online [3]. These power grid benchmarks extracted from real IBM processors which were designed in 90nm/180nm technology node.

A.2 SPICE Netlist Generation

The SPICE Netlist format for the IBM power grid benchmarks are demonstrated here:

- Node name:

  n<net index>_<x location>_<y location>

- Data associated with each layer starts from:

  * layer: <name>,<net>_net: <net index>

  Each layer/net combination is associated with a unique $net - index$.

- Vias starts from:

  * vias from: <net index> to <net index>

  Vias are implemented as resistors or as zero voltage sources.

- Current source:

  iB<block number> <node> 0 <value>
  iB<block number> 0 <node> <value>

  Each current source is split into two components: from V_{DD} to ideal ground and from ideal ground to V_{SS}. Current sources in transient benchmarks are pulse current souces.

- Each circuit file has a global V_{DD} voltage source and each package connection is recognized as a resistor connected to the global source.

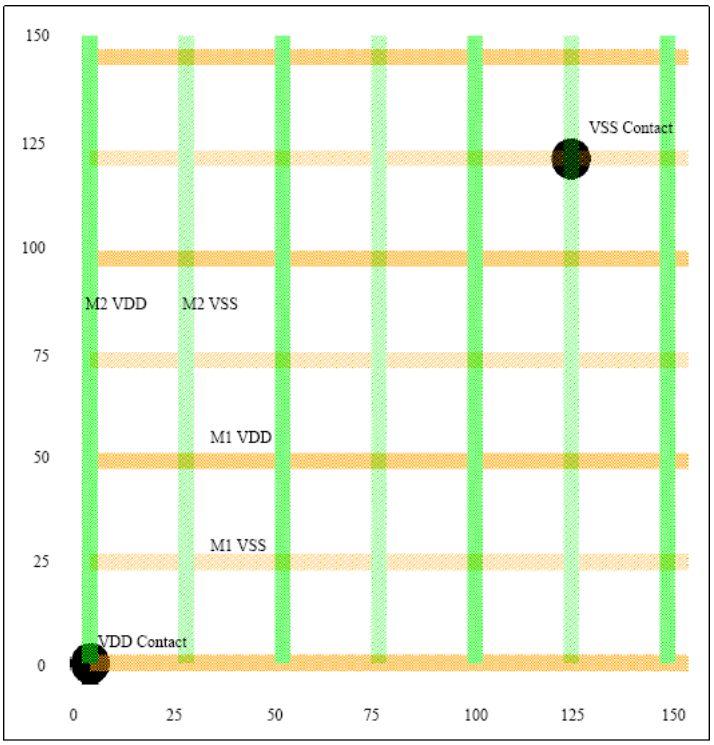


Figure A.1: A small power grid

A.3 Sample IBM Power Grid Benchmark Netlist

A sample IBM Power Grid Benchmark netlist corresponding to Figure A.1 is demonstrated in this section given below.

```
rr0 n3_0_0 _X_n3_0_0 0.5
v1 _X_n3_0_0 0 1
rr2 n2_125_125 _X_n2_125_125 0.5
v3 _X_n2_125_125 0 0
* layer: M1,VDD net: 1
R4 n1_0_0 n1_50_0 1.25
R5 n1_50_0 n1_100_0 1.25
R6 n1_100_0 n1_150_0 1.25
R7 n1_0_50 n1_50_50 1.25
R8 n1_50_50 n1_100_50 1.25
R9 n1_100_50 n1_150_50 1.25
R10 n1_0_100 n1_50_100 1.25
```

```
R11 n1_50_100 n1_100_100 1.25
R12 n1_100_100 n1_150_100 1.25
R13 n1_0_150 n1_50_150 1.25
R14 n1_50_150 n1_100_150 1.25
R15 n1_100_150 n1_150_150 1.25

* vias from: 1 to 3
V16 n1_0_0 n3_0_0 0.0
V17 n1_0_50 n3_0_50 0.0
V18 n1_0_100 n3_0_100 0.0
V19 n1_0_150 n3_0_150 0.0
V20 n1_50_0 n3_50_0 0.0
V21 n1_50_50 n3_50_50 0.0
V22 n1_50_100 n3_50_100 0.0
V23 n1_50_150 n3_50_150 0.0
V24 n1_100_0 n3_100_0 0.0
V25 n1_100_50 n3_100_50 0.0
V26 n1_100_100 n3_100_100 0.0
V27 n1_100_150 n3_100_150 0.0
V28 n1_150_0 n3_150_0 0.0
V29 n1_150_50 n3_150_50 0.0
V30 n1_150_100 n3_150_100 0.0
V31 n1_150_150 n3_150_150 0.0

* layer: M2,VDD net: 3
R32 n3_0_0 n3_0_50 1.25
R33 n3_0_50 n3_0_100 1.25
R34 n3_0_100 n3_0_150 1.25
R35 n3_50_0 n3_50_50 1.25
R36 n3_50_50 n3_50_100 1.25
```

```
R37 n3_50_100 n3_50_150 1.25
R38 n3_100_0 n3_100_50 1.25
R39 n3_100_50 n3_100_100 1.25
R40 n3_100_100 n3_100_150 1.25
R41 n3_150_0 n3_150_50 1.25
R42 n3_150_50 n3_150_100 1.25
R43 n3_150_100 n3_150_150 1.25

* layer: M1,GND net: 0

R44 n0_25_25 n0_75_25 1.25
R45 n0_75_25 n0_125_25 1.25
R46 n0_25_75 n0_75_75 1.25
R47 n0_75_75 n0_125_75 1.25
R48 n0_25_125 n0_75_125 1.25
R49 n0_75_125 n0_125_125 1.25

* layer: M2,GND net: 2

R50 n2_25_25 n2_25_75 1.25
R51 n2_25_75 n2_25_125 1.25
R52 n2_75_25 n2_75_75 1.25
R53 n2_75_75 n2_75_125 1.25
R54 n2_125_25 n2_125_75 1.25
R55 n2_125_75 n2_125_125 1.25

* vias from: 0 to 2

V56 n0_25_25 n2_25_25 0.0
V57 n0_25_75 n2_25_75 0.0
```

```
V58 n0_25_125 n2_25_125 0.0
V59 n0_75_25 n2_75_25 0.0
V60 n0_75_75 n2_75_75 0.0
V61 n0_75_125 n2_75_125 0.0
V62 n0_125_25 n2_125_25 0.0
V63 n0_125_75 n2_125_75 0.0
V64 n0_125_125 n2_125_125 0.0

*

iB0_0_v n1_0_0 0 0.3125m
iB0_0_g 0 n0_25_25 0.3125m
iB0_1_v n1_0_50 0 0.3125m
iB0_1_g 0 n0_25_25 0.3125m
iB0_2_v n1_0_100 0 0.3125m
iB0_2_g 0 n0_25_75 0.3125m
iB0_3_v n1_0_150 0 0.3125m
iB0_3_g 0 n0_25_125 0.3125m
iB0_4_v n1_50_0 0 0.3125m
iB0_4_g 0 n0_25_25 0.3125m
iB0_5_v n1_100_0 0 0.3125m
iB0_5_g 0 n0_75_25 0.3125m
iB0_6_v n1_50_50 0 0.3125m
iB0_6_g 0 n0_25_25 0.3125m
iB0_7_v n1_50_100 0 0.3125m
iB0_7_g 0 n0_25_75 0.3125m
iB0_8_v n1_100_50 0 0.3125m
iB0_8_g 0 n0_75_25 0.3125m
iB0_9_v n1_100_100 0 0.3125m
iB0_9_g 0 n0_75_75 0.3125m
```

```
iB0_10_v n1_50_150 0 0.3125m
iB0_10_g 0 n0_25_125 0.3125m
iB0_11_v n1_100_150 0 0.3125m
iB0_11_g 0 n0_75_125 0.3125m
iB0_12_v n1_150_0 0 0.3125m
iB0_12_g 0 n0_125_25 0.3125m
iB0_13_v n1_150_50 0 0.3125m
iB0_13_g 0 n0_125_25 0.3125m
iB0_14_v n1_150_100 0 0.3125m
iB0_14_g 0 n0_125_75 0.3125m
iB0_15_v n1_150_150 0 0.3125m
iB0_15_g 0 n0_125_125 0.3125m

.op
.end
```

A.4 IBM Power Grid Benchmark Statistics

IBM Power Grid Benchmark Statistics for steady-state analysis are listed in Table A.1. In the table, i for current source, n for nodes (total number, does not take shorts into account), r for resistors (include shorts), s for shorts (zero value resistors and voltage sources) , v for voltage sources (include shorts), l for metal layers. The last column denotes the resistance ranges of the power grid interconnects.

Table A.1: IBM Power Grid Benchmark Statistics for steady-state analysis

Name	**#i**	**#n**	**#r**	**#s**	**#v**	**#l**	resistance ranges
ibmpg1	10774	30638	30027	14208	14308	2	(0,13.38Ω]
ibmpg2	37926	127238	208325	1298	330	5	(0,1.17Ω]
ibmpg3	201054	851584	1401572	461	955	5	(0.9.36Ω]
ibmpg4	276976	953583	1560645	11682	962	6	(0,2.34Ω]
ibmpg5	540800	1079310	1076848	606587	539087	3	(0,1.51Ω]
ibmpg6	761484	1670494	1649002	836107	836239	3	(0.17.16Ω]
ibmpgnew1	357930	1461036	2352355	461	955	NA	(0.17.16Ω]
ibmpgnew2	357930	1461039	1422830	929722	930216	NA	(0,21.6Ω]

A.5 Large Power Grid Benchmarks

Large power grid benchmarks are listed in the following table.

Table A.2: Large power grid benchmark for steady-state analysis

Name	#i	#n	#r	#v
pgckt_10K	9801	10K	19800	200
pgckt_40K	99454	40K	199000	350
pgckt_90K	149599	90K	299000	400
pgckt_250K	248974	250K	499000	1029
pgckt_640K	399654	640K	799000	1029
pgckt_1M	998943	1M	1998000	1059
pgckt_4M	3996980	4M	7996000	3020
pgckt_9M	8994790	9M	17994000	5211
pgckt_16M	15991481	16M	31992000	8523
pgckt_25M	24987989	25M	49990000	12013
pgckt_49M	3499500	49M	6999000	24012

B

Metaheuristics

Contents

B.1 Metaheuristics 147
B.2 Cooperative Coevolution(CC) 147
B.3 NSGA-II based evolutionary algorithm 147

B.1 Metaheuristics

In this appendix, we discuss about the evolutionary computing-based metaheuristics employed in Chapter 4 for design space exploration. Metaheuristics are basically heuristics designed to produce problem-independent solutions of optimization problems. Here, we discuss about Cooperative Coevolution and NSGA-II based metaheuristics which are employed for design space exploration in Chapter 4.

B.2 Cooperative Coevolution(CC)

Cooperative Coevolution is a divide and conquer based approach to solve large scale variable optimization problems. It decomposes a large scale problem into several simple sub-problems. So the basic phenomenon of CC is that it decomposes an n-dimensional decision vector into n subcomponents and then optimizes each of the subcomponents using standard evolutionary optimization algorithm in a round robin fashion. The basic principle of the evolutionary optimization algorithm is to mimic the biological evolution process in generating good candidate solutions for a given objective function. Generally, candidate solutions of an optimization problem play the role of individuals in a population, and these individuals go under reproduction, mutation and recombination and selection to find the optimum solutions for a given objective function. Cooperative coevolution algorithm is stated in Algorithm B.1.

Algorithm B.1: Cooperative Coevolution Algorithm

Input: f, x_{min}, x_{max}, n
Output: Optimized value of f and corresponding variables $x_1, x_2, \cdots, x_n$ *values*
/*grouping based variable decomposition*/;
`groups` ← `grouping(`f, x_{min}, x_{max}, n`);`
/*Optimization stage using evolutionary algorithm*/;
`population` ← `rand(population_size,`n`);;`
for *j* ← *1 to size(groups)* **do**
    `group_num` ← `groups[j];`
    `subpop` ← `population[:,group_num];`
    `subpop` ← `optimizer(best,subpop,FE);`
    `population[:,group_num]` ← `subpop;`
    `(best,best_val)`←`min(population);`

B.3 NSGA-II based evolutionary algorithm

NSGA-II [77] has been employed as it is one of the widely used and first of its kind multi-objective optimization evolutionary algorithm. Evolutionary algorithm solves optimization problems by mim-

icking biological evolution process. It generates a population that undergoes reproduction, mutation, recombination, and selection for many generations corresponding to the objective functions from which we get the optimal solution. In NSGA-II, a random generation of N population is created. Let's called this current generation of population be P'_t. The objective functions are evaluated at all N points of P'_t resulting in P_t. From P_t offspring population Q'_t of size N has been created using crowded tournament selection and polynomial mutation [77].

Similarly, objective functions are evaluated at all points of Q'_t resulting in Q_t. Subsequently, P_t and Q_t are combined to create a combined population R_t of size $2N$. The best of the N population from combined population R_t is selected using non-dominated sorting approach for the next parent population generation P_{t+1}. The entire population R_t are sorted in various nondominated levels, of which each level contains some k number of nondominated population, which is described in Algorithm B.3. These nondominated levels generate nondominated fronts F_i. The fronts are sorted according to their nondomination level. The P_{t+1} population, can be generated by filling the populations from the most nondominated fronts F_i until $|P_{t+1}|+|F_i|\leq N$. However if $|P_{t+1}|+|F_i|> N$ or the number of populations in the first nondominated front is more than N then the front is splitted to have the best diverse solutions in P_{t+1}. For this purpose crowding distance [77, 79] is calculated for all the populations of the last front to be splitted and all $(N-|P_{t+1}|)$ diverse solutions are added after sorting using quick sort in decreasing order of their crowding distance. NSGA-II algorithm is mentioned in Algorithm B.2.

The NSGA-II algorithm, which is used in the earlier section, is mentioned here.

Algorithm B.2: NSGA-II based evolutionary algorithm

Result: Write here the result
Input : Population size N, `maxGeneration`, Problem size
Output: Pareto optimal solutions

Parent population P'_t of size N is generated randomly with the boundary constraint of decision variables.;
$P_t \leftarrow$`evaluate`$(f_i(x))$ at all N points of P_t;
Objective functions are evaluated for parent populations N.;
for $t = 2$ *to* *maxGeneration* **do**
    $Q''_t \leftarrow$ `crowdedTournamentSelection` (P_t);
    $Q'_t \leftarrow$ `polynomialMutation`(Q''_t);
    $Q_t \leftarrow$`evaluate`$(f_i(x))$ at all N points of Q_t;
    P_t and Q_t of size N are combined and created $R_t = P_t \cup Q_t$ of size $2N$.;
    $F_i \leftarrow$`nondominatedSorting`(R_t) where different fronts $F_i,\ i = 1, 2, \cdots$;
    Set new population $P_{t+1} \leftarrow \phi$;
    Set a counter $i \leftarrow 1$;
    foreach F_i **do**
        if $|P_{t+1}|+|F_i| \leq N$ **then**
            $P_{t+1} = P_{t+1} \cup F_i$;
        else
            `CrowdingDistanceAssignment`(F_i);
            $F'_i \leftarrow$ `QuickSortByCrowdingDistanceInDescending`(F_i);
            $P_{t+1} \leftarrow \big(N - |P_{t+1}|\big)$ best solutions of F'_i.
    P_{t+1} is generated.

Algorithm B.3: `nondominatedSorting()`

Input : Combined population $R_t = \{x_1, x_2, \ldots, x_{2N}\}$
Output: Nondominated fronts F_j

Set all nondominated sets $F_j \leftarrow \phi$;
$j \leftarrow 1$;
while $R_t \neq \phi$ **do**
    find the nondominated set F';
    while $i \leq$ **do**
    $F_j \leftarrow F'$ and $R_t = R_t \setminus F'$;
    $j \leftarrow j + 1$;

Bibliography

[1] Y. Zhong and M. Wong, "Fast algorithms for IR drop analysis in large power grid," in *ICCAD*, 2005.

[2] A. B. Kahng, S. Nath, and T. S. Rosing, "On potential design impacts of electromigration awareness," in *2013 18th Asia and South Pacific Design Automation Conference (ASP-DAC)*. IEEE, 2013, pp. 527–532.

[3] S. R. Nassif, "Power grid analysis benchmarks," in *Design Automation Conference, 2008. ASPDAC 2008. Asia and South Pacific*. IEEE, 2008, pp. 376–381. [Online]. Available: https://web.ece.ucsb.edu/~lip/PGBenchmarks/ibmpgbench.html

[4] X. Huang, A. Kteyan, S. X.-D. Tan, and V. Sukharev, "Physics-based electromigration models and full-chip assessment for power grid networks," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 11, pp. 1848–1861, 2016.

[5] S. Chatterjee, V. Sukharev, and F. N. Najm, "Fast physics-based electromigration assessment by efficient solution of linear time-invariant LTI systems," in *Proceedings of the 36th International Conference on Computer-Aided Design*. IEEE Press, 2017, pp. 659–666.

[6] S. Chatterjee, V. Sukharev, and F. N. Najm, "Power grid electromigration checking using physics-based models," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 7, pp. 1317–1330, 2018.

[7] F. N. Najm and V. Sukharev, "Efficient simulation of electromigration damage in large chip power grids using accurate physical models," in *2019 IEEE International Reliability Physics Symposium (IRPS)*. IEEE, 2019, pp. 1–10.

[8] F. N. Najm, "Physical design challenges in the chip power distribution network." in *ISPD*, 2015, p. 101.

[9] X. Huang, Y. Tan, V. Sukharev, and S. X.-D. Tan, "Physics-based electromigration assessment for power grid networks," in *Design Automation Conference (DAC), 2014 51st ACM/EDAC/IEEE*. IEEE, 2014, pp. 1–6.

[10] S. K. Moore, "DARPA picks its first set of winners in electronics resurgence initiative," 2018. [Online]. Available: https://spectrum.ieee.org/tech-talk/semiconductors/design/darpa-picks-its-first-set-of-winners-in-electronics-resurgence-initiative

[11] 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.

[12] 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.

[13] J. R. Black, "Electromigration: A brief survey and some recent results," *IEEE Transactions on Electron Devices*, vol. 16, no. 4, pp. 338–347, 1969.

[14] I. A. Blech, "Electromigration in thin aluminum films on titanium nitride," *Journal of Applied Physics*, vol. 47, no. 4, pp. 1203–1208, 1976.

[15] M. Korhonen, P. Bo/rgesen, K. Tu, and C.-Y. Li, "Stress evolution due to electromigration in confined metal lines," *Journal of Applied Physics*, vol. 73, no. 8, pp. 3790–3799, 1993.

[16] H. Qian, S. R. Nassif, and S. S. Sapatnekar, "Random walks in a supply network," in *Proceedings of the 40th annual Design Automation Conference*. ACM, 2003, pp. 93–98.

[17] H. Qian, S. R. Nassif, and S. S. Sapatnekar, "Power grid analysis using random walks," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 8, pp. 1204–1224, 2005.

[18] B. Boghrati and S. Sapatnekar, "Incremental solution of power grids using random walks," in *Proceedings of the 2010 Asia and South Pacific Design Automation Conference*. IEEE Press, 2010, pp. 757–762.

[19] 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.

[20] H. Qian and S. S. Sapatnekar, "A hybrid linear equation solver and its application in quadratic placement," in *Computer-Aided Design, 2005. ICCAD-2005. IEEE/ACM International Conference on*. IEEE, 2005, pp. 905–909.

[21] E. Chiprout, "Fast flip-chip power grid analysis via locality and grid shells," in *Proceedings of the 2004 IEEE/ACM International conference on Computer-aided design*. IEEE Computer Society, 2004, pp. 485–488.

[22] S. Köse and E. G. Friedman, "Fast algorithms for IR voltage drop analysis exploiting locality," in *Proceedings of the 48th Design Automation Conference*. ACM, 2011, pp. 996–1001.

[23] 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.

[24] Z. Zeng, Z. Feng, P. Li, and V. Sarin, "Locality-driven parallel static analysis for power delivery networks," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 16, no. 3, p. 28, 2011.

[25] L. Zhang and V. Sarin, "Parallel power grid analysis based on enlarged partitions," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 21, no. 2, p. 26, 2016.

[26] S. Chowdhury and M. Breuer, "Minimal area design of power/ground nets having graph topologies," *IEEE Transactions on Circuits and Systems*, vol. 34, no. 12, pp. 1441–1451, 1987.

[27] S. Chowdhury, "Optimum design of reliable IC power networks having general graph topologies," in *Design Automation, 1989. 26th Conference on*. IEEE, 1989, pp. 787–790.

[28] 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.

[29] 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.

[30] X.-D. Tan, C.-J. R. Shi, D. Lungeanu, J.-C. Lee, and L.-P. Yuan, "Reliability-constrained area optimization of VLSI power/ground networks via sequence of linear programmings," in *Proceedings of the 36th annual ACM/IEEE Design Automation Conference*. ACM, 1999, pp. 78–83.

[31] 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.

[32] 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.

[33] S.-D. Tan and C.-J. Shi, "Efficient very large scale integration power/ground network sizing based on equivalent circuit modeling," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 3, pp. 277–284, 2003.

[34] K. Wang and M. Marek-Sadowska, "Power/ground mesh area optimization using multigrid-based technique [IC design]," in *Design, Automation and Test in Europe Conference and Exhibition, 2003*. IEEE, 2003, pp. 850–855.

[35] C. Zhuo, J. Hu, M. Zhao, and K. Chen, "Power grid analysis and optimization using algebraic multigrid," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 4, pp. 738–751, April 2008.

[36] R. Bhooshan, "Novel and efficient IR-drop models for designing power distribution network for sub-100nm integrated circuits," in *Quality Electronic Design, 2007. ISQED'07. 8th International Symposium on.* IEEE, 2007, pp. 287–292.

[37] Z. Zeng and P. Li, "Locality-driven parallel power grid optimization," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 8, pp. 1190–1200, 2009.

[38] R. Jakushokas and E. G. Friedman, "Power network optimization based on link breaking methodology," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 21, no. 5, pp. 983–987, 2013.

[39] H. Zhou, Y. Sun, Z. Sun, H. Zhao, and S. X.-D. Tan, "Electromigration-lifetime constrained power grid optimization considering multi-segment interconnect wires," in *Design Automation Conference (ASP-DAC), 2018 23rd Asia and South Pacific.* IEEE, 2018, pp. 399–404.

[40] S. A. Sadat, M. Canbolat, and S. Köse, "Optimal Allocation of LDOs and Decoupling Capacitors within a Distributed On-Chip Power Grid," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 23, no. 4, p. 49, 2018.

[41] M. Chakraborty, D. Saha, A. Chakrabarti, and S. Bindai, "A CAD approach for pre-layout optimal PDN design and its post-layout verification," *Microprocessors and Microsystems*, vol. 65, pp. 158–168, 2019.

[42] R. Bairamkulov, K. Xu, M. Popovich, J. S. Ochoa, V. Srinivas, and E. G. Friedman, "Power delivery exploration methodology based on constrained optimization," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2019.

[43] C. Decoin, "Power grid analysis - challenges at 20nm and below," *Semiconductor Engineering*, 2014. [Online]. Available: http://semiengineering.com/power-grid-analysis-challenges-at-20nm-and-below/

[44] M. Fawaz and F. N. Najm, "Accurate verification of RC power grids," in *Proceedings of the 2016 Conference on Design, Automation & Test in Europe (DATE).* EDA Consortium, 2016, pp. 814–817.

[45] X. Wang, H. Wang, J. He, S. X.-D. Tan, Y. Cai, and S. Yang, "Physics-based electromigration modeling and assessment for multi-segment interconnects in power grid networks," in *2017 Proceedings of the Conference on Design, Automation & Test in Europe (DATE).* European Design and Automation Association, 2017, pp. 1731–1736.

[46] S. ik Heo, A. B. Kahng, M. Kim, L. Wang, and C. Yang, "Detailed placement for IR drop mitigation by power staple insertion in sub-10nm VLSI," in *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE).* IEEE, 2019, pp. 830–835.

[47] V. Mishra and S. S. Sapatnekar, "Predicting electromigration mortality under temperature and product lifetime specifications," in *Proceedings of the 53rd Annual Design Automation Conference.* ACM, 2016, p. 43.

[48] X. Wang, Y. Yan, J. He, S. X.-D. Tan, C. Cook, and S. Yang, "Fast physics-based electromigration analysis for multi-branch interconnect trees," in *Proceedings of the 36th International Conference on Computer-Aided Design.* IEEE Press, 2017, pp. 169–176.

[49] C. Cook, Z. Sun, E. Demircan, M. D. Shroff, and S. X.-D. Tan, "Fast electromigration stress evolution analysis for interconnect trees using Krylov subspace method," *IEEE Transactions on Very Large Scale Integration VLSI Systems*, vol. 26, no. 5, pp. 969–980, 2018.

[50] V. Sukharev and F. N. Najm, "Electromigration check: Where the design and reliability methodologies meet," *IEEE Transactions on Device and Materials Reliability*, vol. 18, no. 4, pp. 498–507, 2018.

[51] S. M. Nair, R. Bishnoi, M. B. Tahoori, H. Zahedmanesh, K. Croes, K. Garello, G. S. Kar, and F. Catthoor, "Variation-aware physics-based electromigration modeling and experimental calibration for VLSI interconnects," in *2019 IEEE International Reliability Physics Symposium (IRPS).* IEEE, 2019, pp. 1–6.

[52] G. Cui, W. Yu, X. Li, Z. Zeng, and B. Gu, "Machine-learning-driven matrix ordering for power grid analysis," in *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2019, pp. 984–987.

[53] Y.-C. Fang, H.-Y. Lin, M.-Y. Sui, C.-M. Li, and E. J.-W. Fang, "Machine-learning-based dynamic IR drop prediction for ECO," in *2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2018, pp. 1–7.

[54] Y.-C. Liu, C.-Y. Han, S.-Y. Lin, and J. C.-M. Li, "PSN-aware circuit test timing prediction using machine learning," *IET Computers & Digital Techniques*, vol. 11, no. 2, pp. 60–67, 2016.

[55] W.-H. Chang, C.-H. Lin, S.-P. Mu, L.-D. Chen, C.-H. Tsai, Y.-C. Chiu, and M. C.-T. Chao, "Generating routing-driven power distribution networks with machine-learning technique," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 36, no. 8, pp. 1237–1250, 2017.

[56] S.-Y. Lin, Y.-C. Fang, Y.-C. Li, Y.-C. Liu, T.-S. Yang, S.-C. Lin, C.-M. Li, and E. J.-W. Fang, "IR drop prediction of ECO-revised circuits using machine learning," in *2018 IEEE 36th VLSI Test Symposium (VTS)*. IEEE, 2018, pp. 1–6.

[57] F. Ye, F. Firouzi, Y. Yang, K. Chakrabarty, and M. B. Tahoori, "On-chip voltage-droop prediction using support-vector machines," in *2014 IEEE 32nd VLSI Test Symposium (VTS)*. IEEE, 2014, pp. 1–6.

[58] Y. Cao, A. B. Kahng, J. Li, A. Roy, V. Srinivas, and B. Xu, "Learning-based prediction of package power delivery network quality," in *Proceedings of the 24th Asia and South Pacific Design Automation Conference*. ACM, 2019, pp. 160–166.

[59] T. M. et al, "Fast and Memory-Efficient GPU Implementations of Krylov Subspace Methods for Efficient Power Grid Analysis," in *GLSVLSI'13*, May 2âĂŞ3, 2013, Paris, France.

[60] 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.

[61] 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.

[62] 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.

[63] H. Qian and S. S. Sapatnekar, "Hierarchical random-walk algorithms for power grid analysis," *Proceedings of the Asia and South Pacific Design Automation Conference*, 2004.

[64] J. Shi, Y. Cai, W. Hou, L. Ma, S. X. Tan, P. Ho, and X. Wang, "GPU friendly Fast Poisson Solver for Structured Power Grid Network Analysis ," *DAC*, pp. 178–183, 2009.

[65] 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.

[66] R. R. Sarukkai, "Link prediction and path analysis using Markov chains," *Computer Networks*, vol. 33, no. 1, pp. 377–386, 2000.

[67] W. Torrez, "The birth and death chain in a random environment : Instability and extinction theorems," *The Annals of Probability*, vol. 6, 1978.

[68] A. V. Chechkin, R. Metzler, J. Klafter, and V. Y. Gonchar, "Introduction to the Theory of Lévy Flights," *Anomalous Transport: Foundations and Applications*, no. 1, pp. 129–162, 2008.

[69] S. Kose and E. G. Friedman, "Effective resistance of a two layer mesh," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 58, no. 11, pp. 739–743, 2011.

[70] *CUDA Zone, NVIDIA Developer*. [Online]. Available: https://developer.nvidia.com/cuda-zone

[71] H. Qian and S. S. Sapatnekar, "A hybrid linear equation solver and its application in quadratic placement," in *Computer-Aided Design, 2005. ICCAD-2005. IEEE/ACM International Conference on*, nov 2005, pp. 905–909.

[72] I. P. Vaisband, R. Jakushokas, M. Popovich, A. V. Mezhiba, S. Köse, and E. G. Friedman, *On-Chip Power Delivery and Management.* Springer, 2016.

[73] E. Salman and E. Friedman, *High Performance Integrated Circuit Design.* McGraw Hill Professional, 2012.

[74] Q. K. Zhu, *Power distribution network design for VLSI.* John Wiley & Sons, 2004, chapter 4.

[75] E. T. Ogawa, K.-D. Lee, V. A. Blaschke, and P. S. Ho, "Electromigration reliability issues in dual-damascene Cu interconnections," *IEEE Transactions on reliability*, vol. 51, no. 4, pp. 403–419, 2002.

[76] P. S. Ho, G. Wang, M. Ding, J.-H. Zhao, and X. Dai, "Reliability issues for flip-chip packages," *Microelectronics Reliability*, vol. 44, no. 5, pp. 719–737, 2004.

[77] 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.

[78] K. Miettinen, *Nonlinear multiobjective optimization.* Springer Science & Business Media, 2012, vol. 12.

[79] K. Deb, *Multi-objective optimization using evolutionary algorithms.* John Wiley & Sons, 2001, vol. 16.

[80] Z. Yang, K. Tang, and X. Yao, "Large scale evolutionary optimization using cooperative coevolution," *Information Sciences*, vol. 178, no. 15, pp. 2985–2999, 2008.

[81] M. A. Potter and K. A. De Jong, "A cooperative coevolutionary approach to function optimization," in *International Conference on Parallel Problem Solving from Nature.* Springer, 1994, pp. 249–257.

[82] Y. Liu, X. Yao, Q. Zhao, and T. Higuchi, "Scaling up fast evolutionary programming with cooperative coevolution," in *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*, vol. 2. IEEE, 2001, pp. 1101–1108.

[83] F. Van den Bergh and A. P. Engelbrecht, "A cooperative approach to particle swarm optimization," *IEEE transactions on evolutionary computation*, vol. 8, no. 3, pp. 225–239, 2004.

[84] Y.-j. Shi, H.-f. Teng, and Z.-q. Li, "Cooperative co-evolutionary differential evolution for function optimization," *Advances in natural computation*, pp. 428–428, 2005.

[85] Z. Yang, K. Tang, and X. Yao, "Self-adaptive differential evolution with neighborhood search," in *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on.* IEEE, 2008, pp. 1110–1116.

[86] M. N. Omidvar, X. Li, Z. Yang, and X. Yao, "Cooperative co-evolution for large scale optimization through more frequent random grouping," in *Evolutionary Computation (CEC), 2010 IEEE Congress on.* IEEE, 2010, pp. 1–8.

[87] 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.

[88] A. Saha and T. Ray, "Equality constrained multi-objective optimization," in *2012 IEEE Congress on Evolutionary Computation.* IEEE, 2012, pp. 1–7.

[89] L. M. Antonio and C. A. C. Coello, "Use of cooperative coevolution for solving large scale multiobjective optimization problems," in *Evolutionary Computation (CEC), 2013 IEEE Congress on.* IEEE, 2013, pp. 2758–2765.

[90] "The gold standard for accurate circuit simulation," Online accessed December 2017. [Online]. Available: https://www.synopsys.com/verification/ams-verification/circuit-simulation/hspice.html

[91] J.-P. Li, M. E. Balazs, G. T. Parks, and P. J. Clarkson, "A species conserving genetic algorithm for multimodal function optimization," *Evolutionary computation*, vol. 10, no. 3, pp. 207–234, 2002.

[92] A. Ng, "Cs229 lecture notes, deep learning." [Online]. Available: http://cs229.stanford.edu/notes/cs229-notes-deep_learning.pdf

[93] D. P. Kingma and J. Ba, "Adam : A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[94] B. Yu, X. Xu, S. Roy, Y. Lin, J. Ou, and D. Z. Pan, "Design for manufacturability and reliability in extreme-scaling VLSI," *Science China Information Sciences*, vol. 59, no. 6, p. 061406, 2016.

[95] S. X.-D. Tan, H. Amrouch, T. Kim, Z. Sun, C. Cook, and J. Henkel, "Recent advances in EM and BTI induced reliability modeling, analysis and optimization," *Integration*, vol. 60, pp. 132–152, 2018.

[96] B. Bailey, "Chip aging becomes design problem," 2018. [Online]. Available: https://semiengineering.com/chip-aging-becomes-design-problem/

[97] Y.-J. Huang, C.-L. Pan, S.-C. Lin, and M.-H. Guo, "Machine-learning approach in detection and classification for defects in TSV-based 3-D IC," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 8, no. 4, pp. 699–706, 2018.

[98] I. A. M. Elfadel, D. S. Boning, and X. Li, "Machine learning in VLSI computer-aided design," 2018.

[99] S. Chatterjee, M. Fawaz, and F. N. Najm, "Redundancy-aware power grid electromigration checking under workload uncertainties," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 9, pp. 1509–1522, 2015.

[100] S. Köse and E. G. Friedman, "Efficient algorithms for fast IR drop analysis exploiting locality," *Integration*, vol. 45, no. 2, pp. 149–161, 2012.

[101] P. Inna, R. Jakushokas, M. Popovich, A. V. Mezhiba, S. Köse, E. G. Friedman *et al.*, "Closed-form expressions for fast IR drop analysis," in *On-Chip Power Delivery and Management*. Springer, 2016, pp. 373–396.

[102] K.-D. Lee, "Electromigration recovery and short lead effect under bipolar-and unipolar-pulse current," in *Reliability Physics Symposium (IRPS), 2012 IEEE International*. IEEE, 2012, pp. 6B–3.

[103] K. P. Murphy, *Machine learning: a probabilistic perspective*, Cambridge, MA, 2012.

[104] Y. Zhong and M. D. Wong, "Thermal-aware IR drop analysis in large power grid," in *9th International Symposium on Quality Electronic Design (isqed 2008)*. IEEE, 2008, pp. 194–199.

[105] C. L. Notes, "Deep learning," Online accessed June 2019. [Online]. Available: http://cs229.stanford.edu/notes/cs229-notes-deep_learning.pdf

[106] V. Sukharev, X. Huang, H.-B. Chen, and S. X.-D. Tan, "IR-drop based electromigration assessment: parametric failure chip-scale analysis," in *Proceedings of the 2014 IEEE/ACM International Conference on Computer-Aided Design*. IEEE Press, 2014, pp. 428–433.

[107] "Scikit-learn - machine learning in python," Online accessed June 2019. [Online]. Available: https://scikit-learn.org/stable/

[108] T. K. Ho, "Random decision forests," in *Document analysis and recognition, 1995., proceedings of the third international conference on*, vol. 1. IEEE, 1995, pp. 278–282.

[109] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, ser. Springer series in statistics. Springer, 2009.

[110] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," *ACM transactions on intelligent systems and technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[111] C. E. Rasmussen, "Gaussian processes in machine learning," in *Advanced lectures on machine learning*. Springer, 2004, pp. 63–71.

[112] S. R. Nassif, "Power grid analysis benchmarks," in *Proceedings of the 2008 Asia and South Pacific Design Automation Conference*. IEEE Computer Society Press, 2008, pp. 376–381.

Publications Related to Thesis

Journal Publications

1. **Sukanta Dey**, Sukumar Nandi and Gaurav Trivedi, **"PGOpt: Multi-objective Design Space Exploration Framework for Large-Scale On-Chip Power Grid Design in VLSI SoC using Evolutionary Computing Technique "** in *Elsevier Microprocessor and Microsystems Journal (MICPRO), Vol. 81, Article 103440, March 2021.*

2. **Sukanta Dey**, Sukumar Nandi and Gaurav Trivedi, **"Machine Learning Approach for Fast Electromigration Aware Aging Prediction in Incremental Design of Large Scale On-Chip Power Grid Network "** in *ACM Transactions on Design Automation of Electronic Systems (TODAES), Vol 25, Issue 5, Article 42, July 2020.*

3. **Sukanta Dey**, Sukumar Nandi and Gaurav Trivedi, **"Robust Deep Learning Approach for On-Chip Power Grid Design"** *(Under preparation).*

Conference Proceedings:

1. **Sukanta Dey**, Sukumar Nandi and Gaurav Trivedi, **"PowerPlanningDL: Reliability-Aware Framework for On-Chip Power Grid Design using Deep Learning "**, in *IEEE/ACM Design, Automation and Test in Europe (DATE 2020), Grenoble, France, 9-13th March 2020. (long presentation paper).*

2. **Sukanta Dey**, Satyabrata Dash, Sukumar Nandi and Gaurav Trivedi, **"PGIREM: Reliability-Constrained IR Drop Minimization and Electromigration Assessment of VLSI Power Grid Networks using Cooperative Coevolution"**, in *IEEE Computer Society Annual Symposium on VLSI (ISVLSI 2018), Hong Kong SAR, China, 9-11th July 2018.*

3. **Sukanta Dey**, Satyabrata Dash, Sukumar Nandi and Gaurav Trivedi, **"Markov Chain Model using Levy Flight for VLSI Power Grid Analysis"**, in 30^{th} *International Conference on VLSI Design, (VLSID 2017), Hyderabad, India, 7-11th Jan 2017.*

Book Chapters:

1. **Sukanta Dey**, Sukumar Nandi and Gaurav Trivedi, **"PGRDP: Reliability, Delay, and Power-Aware Area Minimization of Large-Scale VLSI Power Grid Network using Cooperative Coevolution"**, in *in the Book: Intelligent Computing Paradigm - Recent Trends, Springer (Chapter 6), 2019.*

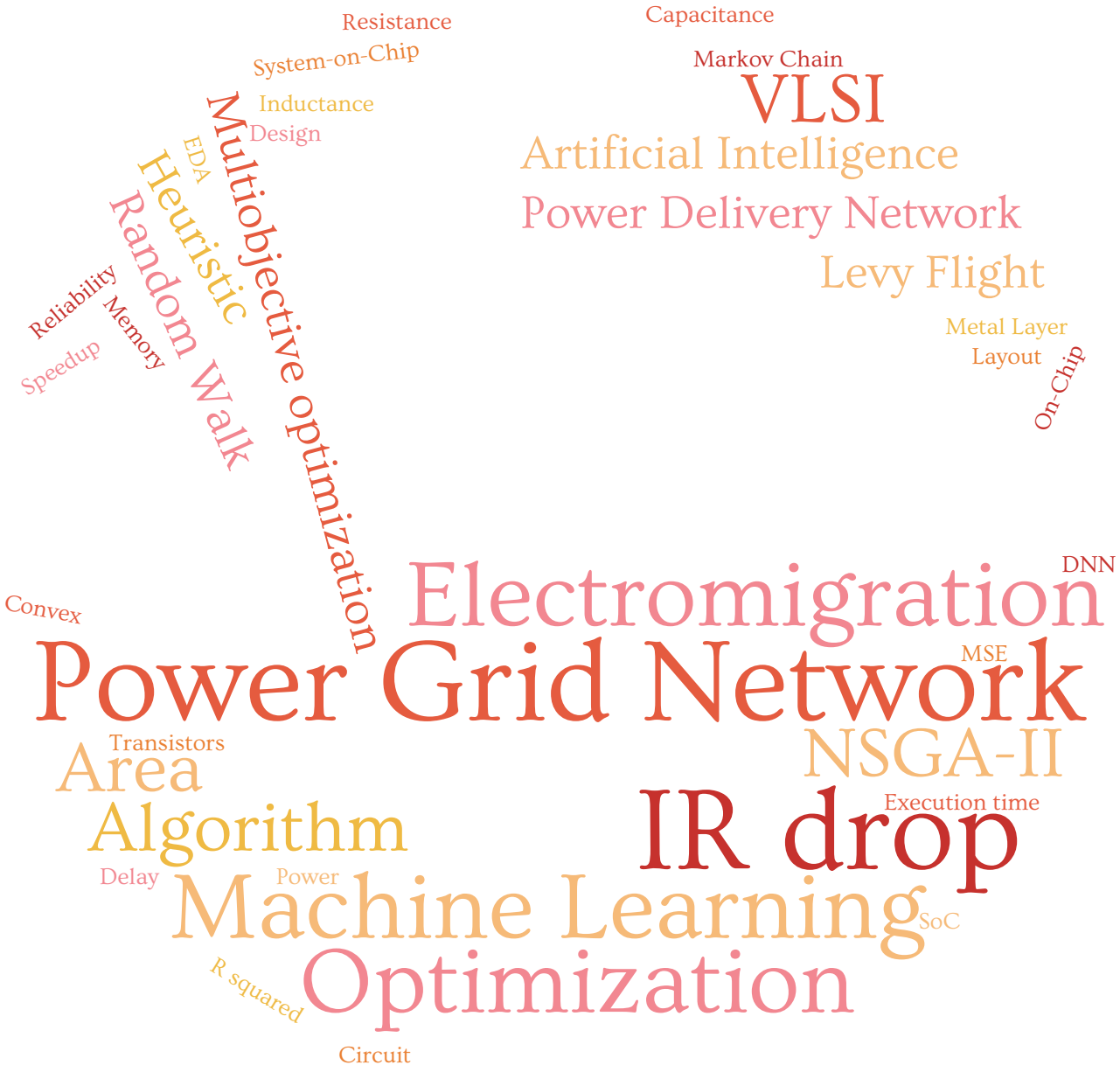
Other Publications (Thesis Related)

1. **Sukanta Dey**, Sukumar Nandi and Gaurav Trivedi, **"Thermal-Aware Management for Reliability Enhancement of Power/Ground TSV in 3D ICs"** *(To be submitted to Journal).*

2. **Sukanta Dey**, Sukumar Nandi and Gaurav Trivedi, **"ETDA: Exploiting Training Data for Executing IR Drop Attack in Learning-based VLSI Power Planner "**, *(To be submitted to Conference Proceedings).*

Author: Sukanta Dey

Sukanta Dey is pursuing his Dual Degree (M.Tech. + Ph.D.) in the Department of Computer Science & Engineering at Indian Institute of Technology Guwahati, India, since July 2014. His Ph.D. thesis focuses on Machine Learning techniques for VLSI CAD. He received his B.E. degree in Electronics & Telecommunication Engineering from Assam Engineering College, India, in July 2014. He has done an internship at Texas Instruments India, Bengaluru, India, for a project related to VLSI Layout Design Automation in December 2015. His research interests include VLSI CAD, VLSI Physical Design, Reliability Issues of SoC Design & interconnects, Machine Learning, and Evolutionary Computation.


Resistance
System-on-Chip
Inductance
Design
EDA
Multiobjective optimization
Heuristic
Random Walk
Reliability
Memory
Speedup
Capacitance
Markov Chain
VLSI
Artificial Intelligence
Power Delivery Network
Levy Flight
Metal Layer
Layout
On-Chip
DNN
Electromigration
Convex
Power Grid Network
MSE
Transistors
NSGA-II
Area
Execution time
Algorithm
IR drop
Delay
Power
Machine Learning
SoC
R squared
Optimization
Circuit