Optimization of Circuit and System

A Project Report Submitted by

Suraj Prajapati

in partial fulfillment of the requirements for the award of the degree of

M.Tech

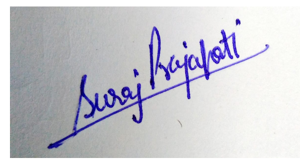
Indian Institute of Technology Jodhpur

Department of Electrical Engineering

April, 2025

Declaration

I hereby declare that the work presented in this Project Report titled Optimization of Circuit and System submitted to the Indian Institute of Technology Jodhpur in partial fulfilment of the requirements for the award of the degree of M.Tech, is a bonafide record of the research work carried out under the supervision of **Dr. Jai Narayan Tripathi** . The contents of this Project Report in full or in parts, have not been submitted to, and will not be submitted by me to, any other Institute or University in India or abroad for the award of any degree or diploma.



Signature

Suraj Prajapati

M23EEV019

Certificate

This is to certify that the Project Report titled Title of the Project Report, submitted by Suraj Prajapati (M23EEV019) to the Indian Institute of Technology Jodhpur for the award of the degree of M.Tech, is a bonafide record of the research work done by him under my supervision. To the best of my knowledge, the contents of this report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Signature
Dr. Jai Narayan Tripathi

Acknowledgements

I would like to express my heartfelt gratitude to **Dr. Jai Narayan Tripathi** , my project supervisor, for his unwavering support, guidance, and invaluable insights throughout the duration of this project. His expertise and mentorship have been instrumental in shaping the course of my research, and I am deeply thankful for his dedication to my academic and professional growth.

Abstract

Maintaining power integrity inside the power delivery networks (PDNs) of high-speed VLSI systems has become extremely difficult due to Moore's Law-driven increases in integrated circuit (IC) complexity. Simultaneous switching noise (SSN) and non-optimal impedance profiles can contribute to power supply noise (PSN), which can impair performance and call for efficient optimization. Decoupling capacitors, often known as decaps, are essential for lowering impedance, but choosing and positioning them effectively involves resolving challenging combinatorial issues. In order to optimize decap placement in PDNs, this report uses **Particle Swarm Optimization (PSO)**, which effectively minimizes impedance below the goal threshold with little computational effort. In order to further improve optimization efficiency, it also examines the theoretical underpinnings of the **Radial Basis Function Network (RBFN)-Based Surrogate-Assisted Swarm Intelligence Approach**, which combines surrogate models with PSO to approximate computationally costly objective functions.

The report provides case examples, theoretical insights, and algorithmic specifics that show how well PSO works in real-world PDN optimization. The investigation of RBFN-based methods demonstrates their potential for incorporation in the future, providing notable accuracy gains at runtime. This study emphasizes how metaheuristic and surrogate-assisted approaches can revolutionize the way power integrity issues in contemporary devices are handled.

Contents

List of Figures

Optimization of Circuit and System

1 Introduction and background

Very-large-scale integration (VLSI) systems that are extremely complex and fast have been made possible by the quick scaling of transistor technology, which is fueled by Moore's Law. However, preserving power integrity—a crucial need for the dependable operation of integrated circuits (ICs)—has become much more difficult as a result of this advancement. When it comes to providing low-noise power to the active components of integrated circuits, power delivery networks (PDNs) are essential. Power supply noise (PSN), which can impair system performance and dependability, is frequently the result of non-ideal impedance profiles in PDNs that are brought on by parasitic effects and simultaneous switching noise (SSN). Decoupling capacitors, often known as decaps, are used to mitigate PSN and lower impedance in order to address these problems. However, because there are several placement ports and a large number of capacitor alternatives, choosing and positioning decaps properly is a difficult combinatorial task.

Conventional techniques for decap placement optimization [1]frequently depend on thorough assessments, which makes them computationally costly and ineffective, particularly for large-scale PDNs. Because of these drawbacks, metaheuristic optimization methods—like Particle Swarm Optimization (PSO)—have become useful instruments for resolving challenging optimization issues. By directing possible solutions, or particles, using both global and personal best values, PSO effectively searches the solution space, drawing inspiration from the behavior of natural swarms. Furthermore, cutting-edge strategies like Radial Basis Function Network (RBFN)-Based Surrogate-Assisted Optimization present a viable way to increase computing efficiency even more. By approximating costly objective functions, RBFNs greatly enhance runtime efficiency and eliminate the need for frequent real evaluations.

Through thorough case studies, this research emphasizes the use of PSO to optimize decap location and lower PDN impedance. For possible future integration, it also offers a thorough examination of the computational concepts and theoretical underpinnings of RBFN-based surrogate-assisted optimization. These methods highlight the revolutionary potential of fusing metaheuristic algorithms with surrogate modeling techniques to address important power integrity issues in contemporary electronics. .

2 Problem definition and Objective

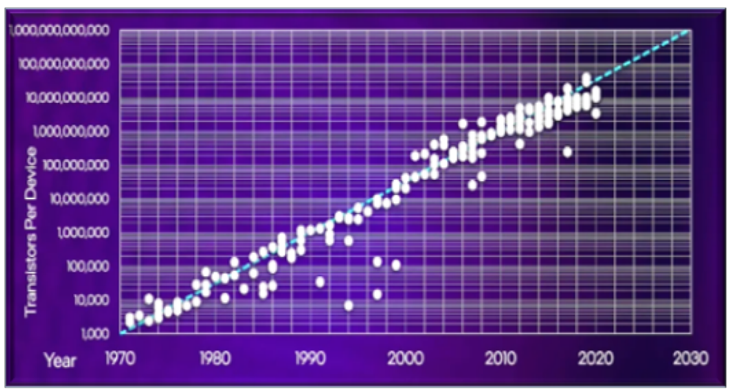
In high-speed VLSI systems, power delivery network (PDN) optimization is essential to guarantee a steady power supply and reduce power supply noise (PSN) brought on by simultaneous switching noise (SSN) and impedance changes. Low impedance is achieved by using decoupling capacitors, or decaps. However, because there are so many different combinations, choosing and positioning them optimally is a difficult combinatorial problem. This work's goal is to use the fewest decaps possible to effectively reduce PDN impedance below a predetermined target threshold. Particle Swarm Optimization (PSO), a metaheuristic approach that lowers computing overhead in comparison to exhaustive approaches, is used to do this. In order to further speed up the optimization process without sacrificing accuracy and open the door for workable solutions to power integrity issues in contemporary electronics, the use of surrogate-assisted models—such as Radial Basis Function Networks (RBFNs)—is also investigated.

3 Theory

Optimizing the power distribution network (PDN) is essential to preserving power integrity in VLSI systems. Particle Swarm Optimization (PSO) has been adopted for effective decoupling capacitor placement because traditional methods are computationally demanding. Radial Basis Function Networks (RBFNs) have been investigated as a surrogate-assisted method to further increase efficiency, lowering runtime without sacrificing accuracy. For future developments, this research looks at RBFN integration and uses PSO.

3.1 Introduction

The development of contemporary electronics has been fueled by the Moore's Law-predicted constant scaling of transistor size. According to Moore's Law, an integrated circuit's (IC) transistor count doubles roughly every two years, allowing for increasingly sophisticated systems and exponential increases in processing power. Transistor density in integrated circuits has expanded dramatically as a result of this trend, which has serious ramifications for system performance and power integrity.



[2]

Figure 3.1: Illustration of Moore's Law

3.1.1 Transistor Scaling's Effect

The current needed for these circuits rises in proportion to the transistors' increasing density and decreasing size. High-speed very-large-scale integration (VLSI) systems are more vulnerable to power supply noise (PSN) and other performance-degrading effects because of the reduced noise margins caused by this higher current demand. Challenges like higher power dissipation, problems with thermal management, and increased sensitivity to simultaneous switching noise (SSN) and other electrical noise sources are also brought on by the size reduction.

3.1.2 Problems with Power Integrity

The capacity of a system to supply all of its components with clean, steady power is known as power integrity. It gets harder to keep a steady power supply for high-frequency integrated circuits as transistor density increases. In order to keep noise levels and voltage fluctuations within tolerable bounds and preserve system functionality, power delivery networks (PDNs) are essential for handling these difficulties. However, parasitic effects and simultaneous switching lead to PDNs' non-ideal impedance profiles[3] which frequently create power supply fluctuations that impair IC performance.

Decoupling capacitors, often known as decaps, are frequently used to reduce impedance and reduce noise in order to address these problems.[4] By lowering the anti-resonance peaks in the PDN impedance profile, these capacitors provide steady power delivery. However, because there are so many capacitors and placement ports available, choosing and positioning decaps is not simple and presents a significant combinatorial optimization challenge.

3.2 Noises

High-speed VLSI systems' performance and dependability are greatly impacted by noise in power delivery networks (PDNs), which results in power supply instability and voltage fluctuations. For integrated circuits (ICs) to remain efficient and guarantee signal integrity, effective noise management is essential. Simultaneous switching noise (SSN) and power supply noise (PSN) are the two main forms of noise that impact PDNs.

3.2.1 Power Supply Noise (PSN)

Unwanted voltage fluctuations in power delivery networks (PDNs) that affect integrated circuit (IC) performance are referred to as power supply noise (PSN) [5]. PSN results from power supply oscillations brought on by PDN components' less-than-ideal behavior, such as parasitics and switching activity.
Sources:

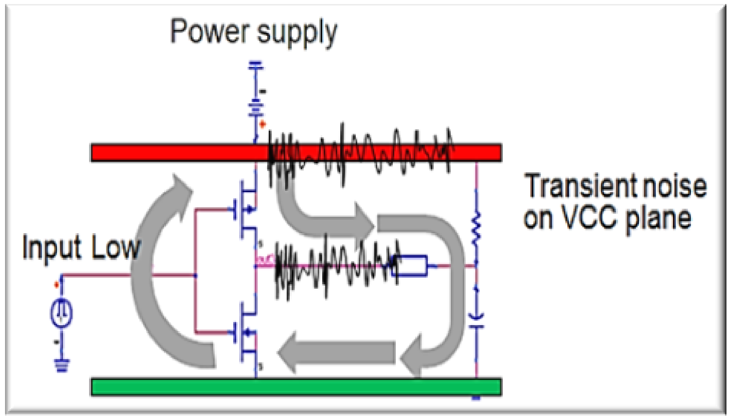
• Switching Activity: PSN is facilitated by the transient currents produced by millions of transistors operating simultaneously.

• External electromagnetic disturbances that couple with the PDN are known as electromagnetic interference, or EMI.

• Parasitic Elements: Noise and changes in impedance are caused by inductive and resistive parasitics in the PDN.

Impacts: • Problems with Signal Integrity: Variations in voltage might cause problems in data transmission.

• Performance Degradation: High-speed systems become less dependable and efficient when noise levels rise.



[5]

Figure 3.2: Power Supply Noise

3.2.2 Simultaneous Switching Noise (SSN)

When several outputs of a digital circuit transition states at the same time, it results in Simultaneous Switching Noise (SSN),[6] sometimes referred to as "ground bounce," which causes noticeable voltage variations on the ground and power planes. Reasons:

• Simultaneous Switching Output (SSO): When several gates change at once, a high current demand occurs.

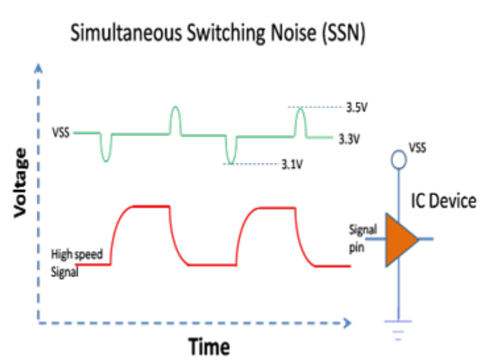
• Inductive Effects: When current changes quickly, parasitic inductance in the PDN results in voltage spikes.

• Current Demand: Noise is amplified by high current transients during switching.

Effects include: • Voltage Fluctuations: Stable power transmission is disrupted by significant voltage level spikes or dips.

• Problems with Signal Integrity: Variability impairs system dependability by interfering with timing and data processing.

In high-speed VLSI systems, PSN and SSN provide significant issues that call for sophisticated strategies like impedance-controlled PDN design and the ideal positioning of decoupling capacitors to lessen their impact.

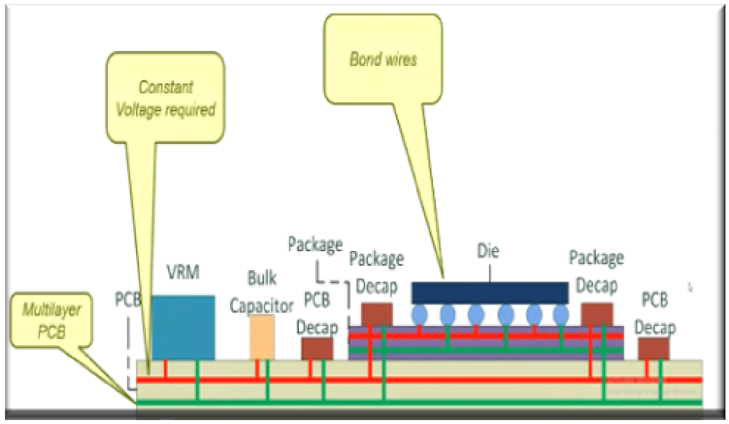


[6]

Figure 3.3: Simultaneous Switching Noise

3.3 PDNs (Power Delivery Networks)

Integrated circuit (IC) components receive consistent, low-noise DC power from Power Delivery Networks (PDNs)[7]. They are made up of voltage regulators, printed circuit boards (PCBs), packaging, and interconnects. A well-designed PDN lowers power supply noise (PSN) and guarantees signal integrity in high-speed VLSI systems. However, PDN components' less-than-ideal properties frequently cause power supply variations, which presents serious difficulties for power integrity experts.



[8]

Figure 3.4: Power Delivery Network (PDN)

3.3.1 PDN Impedance

PDN impedance plays a crucial role in defining the network's performance and dependability. Among the significant characteristics are:

- **PDN Ratio:** The ratio of the actual PDN impedance (Z_{pdn}) to the target impedance (Z_t). A PDN ratio less than 1 indicates a minimal likelihood of noise-related issues.

- **Target Impedance (Z_t):** The target impedance is defined as:
  $Z_t = \frac{\Delta V_n}{I_{max-t}}$
  $where \Delta V_n$ is the maximum allowable voltage noise, and I_{max-t} is the worst-case transient current. Lower impedance ensures better power integrity.

- **Low Impedance Cost:** Achieving low impedance often requires the use of additional components, PCB layers, or expensive materials, increasing the cost and complexity of the design.

- **Role of Decoupling Capacitors (Decaps):** Decaps enhance power integrity by reducing anti-resonance peaks in the impedance profile.[9] They are strategically positioned to ensure efficient noise suppression and minimize self-impedance.

- **Difficulties with Decap Selection:** The large number of port and capacitor combinations makes decap placement optimization challenging. Computationally efficient techniques, such as Particle Swarm Optimization (PSO), are necessary to address this problem.

4 Description of system

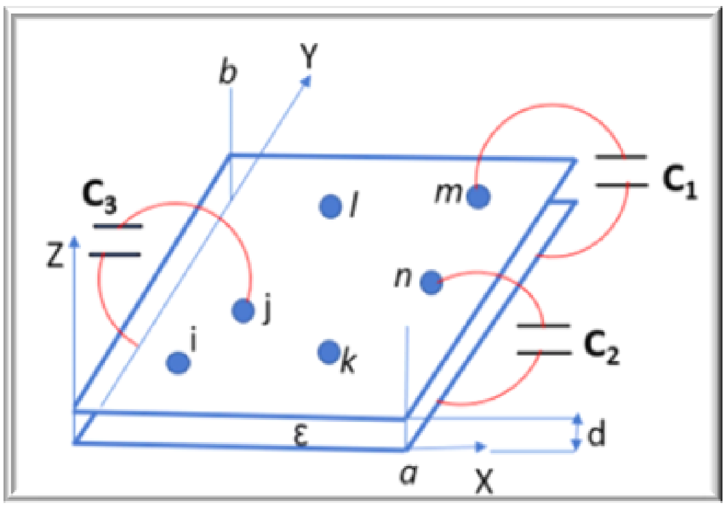
To reduce impedance and ensure steady power delivery in high-speed VLSI systems, this system focuses on strategically placing decoupling capacitors, or decaps, within power delivery networks (PDNs).

4.1 Placement of Decaps

Decoupling capacitors are placed on power planes paired with ground planes to suppress noise and reduce the anti-resonance peaks in the impedance profile. During transient events, these capacitors act as local energy reservoirs, guaranteeing a steady supply of power to IC components.

4.2 Port-Based Connection

Decaps are connected to power and ground planes through specified ports on the board or package configuration. To achieve optimal impedance reduction, these ports must be strategically positioned.



[10]

Figure 4.1: Connection of Decaps on port

4.3 Calculating Equivalent Self-Impedance

The equivalent impedance of the PDN, incorporating decaps (Z_{eq}), is calculated using the formula:

$$Z_{eq} = \left(Z_{pdn}^{-1} + Z_{decap}^{-1}\right)^{-1}$$

where Z_{pdn} represents the intrinsic impedance of the PDN, and Z_{decap} represents the impedance contributed by the decaps. This computation helps assess the effectiveness of decaps in enhancing power integrity.

4.4 Goal of Optimization

The primary goal is to minimize the maximum self-impedance (Z_{obj}) of the PDN, ensuring it remains below the intended impedance threshold. The optimization aims to reduce noise and improve stability across all operating frequencies.

4.5 Objective Function

The system's optimization objective is mathematically expressed as:

$$Z_{obj} = \max\left(Z_{eq}(i, j, f)\right)$$

where i and j represent the port number and decap index, respectively, and f represents the frequency.

4.6 Objective

The ultimate goal is to effectively install the minimum number of decaps to lower the PDN's self-impedance below the specified threshold, achieving both cost-effectiveness and power integrity. By strategically placing decaps and employing optimization techniques, the system ensures robust performance under high-speed operating conditions.

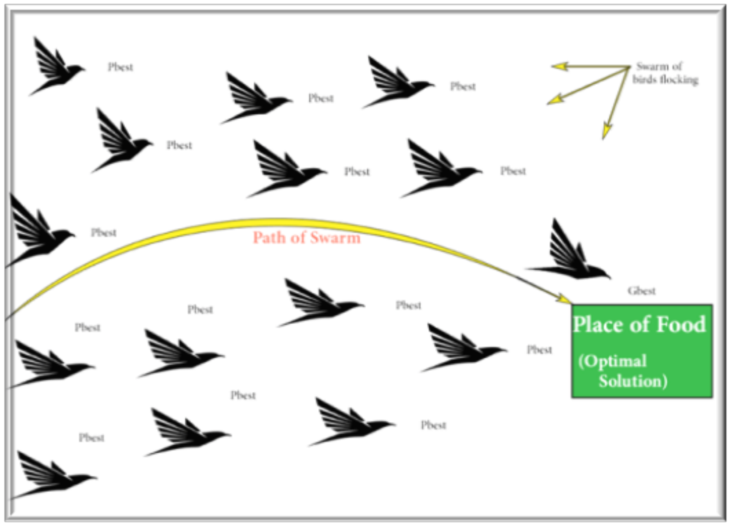
This methodology addresses the complexities of PDN design by integrating decap placement, impedance calculations, and optimization into a cohesive framework, as illustrated in the accompanying diagram of decap placement across multiple ports.

5 Optimization Technique

Large-scale combinatorial issues, such as decap optimization, can be successfully solved via computational intelligence-based methods. Numerous strategies have been investigated, such as metaheuristic techniques and machine learning[9] (e.g., ANNs, reinforcement learning). Although ANNs can approximate objective functions, they are ineffective for genuine evaluations, computationally demanding, and prone to errors. For complicated, large-scale issues, metaheuristic approaches such as Particle Swarm Optimization (PSO) can be time-consuming, despite their shown effectiveness. Surrogate-Assisted PSO (SuA-PSO) incorporates surrogate models, such as Radial Basis Function Networks (RBFNs), to indirectly approximate the objective function in order to overcome these constraints. By carefully choosing which particles to evaluate, these models greatly speed up optimization by lowering the number of actual function evaluations. SuA-PSO is a quicker and more effective method than traditional techniques since RBFNs are favored for their low computational cost, excellent accuracy, and robustness with smaller datasets.

5.1 Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO), introduced by Kennedy and Eberhart [11]in 1995, is a population-based optimization algorithm inspired by the collective behavior of birds flocking or fish schooling. It is popular for its simplicity and effectiveness in solving complex optimization problems. PSO belongs to the class of swarm intelligence algorithms.



[12]

Figure 5.1: PSO Analogy

5.1.1 Overview

PSO utilizes a "swarm" of particles, each representing a potential solution in the search space. The particles move through the space, guided by two main factors:

- **Personal Best (p_{best}):** Each particle's own best-known position.
- **Global Best (g_{best}):** The best-known position among all particles.

The objective is to minimize or maximize the objective function and find the optimal solution.

5.1.2 Equations for Particle Movement

The movement of particles is determined by updates to their velocity (V) and position (X) as follows:

Velocity Update:

$$V_i^{t+1} = \omega V_i^t + c_1 r_1 (p_{best,i} - X_i^t) + c_2 r_2 (g_{best} - X_i^t)$$

- ω: Inertia weight, regulates the balance between exploration and exploitation.
- c_1, c_2: Acceleration coefficients representing cognitive (individual) and social (swarm) influences.
- r_1, r_2: Random numbers in $[0, 1]$, introducing stochasticity to the movement.

Position Update:

$$X_i^{t+1} = X_i^t + V_i^{t+1}$$

5.1.3 Key Parameters

- **Inertia Weight (ω):** Controls the influence of a particle's previous velocity. A high ω encourages exploration (global search), while a low ω focuses on exploitation (local search). A common strategy is the Linear Decreasing Inertia Weight (LDIW)[13], calculated as:

  $$\omega = \omega_{max} - \frac{(\omega_{max} - \omega_{min}) \cdot t}{T_{max}}$$

  where t is the current iteration, and T_{max} is the total number of iterations.

- **Cognitive Coefficient (c_1):** Encourages particles to move toward their personal best positions.

- **Social Coefficient (c_2):** Encourages particles to follow the swarm's global best position.

5.1.4 Exploration vs. Exploitation

- **Exploration:** Ensures diversity by spreading particles throughout the search space. Controlled by a high inertia weight (ω).

- **Exploitation:** Refines the search as particles converge toward promising regions. Achieved by adjusting c_1 and c_2 to emphasize the best-known solutions.

5.1.5 Advantages of PSO

- Easy to implement and computationally efficient.

- Suitable for non-differentiable objective functions as it does not require gradient information.

- Adaptable for balancing global and local searches through parameter tuning.

5.1.6 PSO Algorithm

1. **Initialization:**

   - Initialize a swarm of particles with random positions (X_i) and velocities (V_i) in the search space.

   - Assign random values to personal best ($p_{best,i}$) for each particle and global best (g_{best}) for the swarm.

   - Define parameters: maximum number of iterations (T_{max}), inertia weight (ω), cognitive coefficient (c_1), and social coefficient (c_2).

2. **Evaluate Fitness:** Compute the fitness of each particle using the objective function. Update $p_{best,i}$ and g_{best} based on fitness values.

3. **Update Velocity:** Use the velocity update equation:

$$V_i^{t+1} = \omega V_i^t + c_1 r_1 (p_{best,i} - X_i^t) + c_2 r_2 (g_{best} - X_i^t)$$

4. **Update Position:** Use the position update equation:

$$X_i^{t+1} = X_i^t + V_i^{t+1}$$

5. **Check Stopping Criteria:** Stop if the maximum number of iterations (T_{max}) is reached or the desired fitness is achieved. Otherwise, return to Step 2.

6. **Output:** Return the global best position (g_{best}) as the optimal solution.

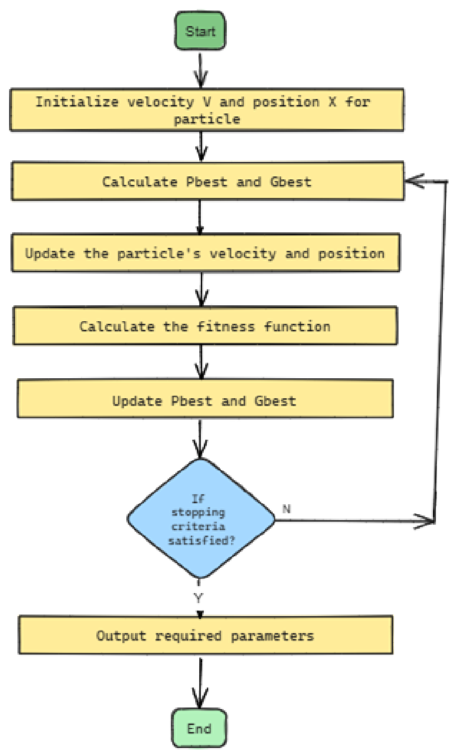


Figure 5.2: Flowchart of PSO Algorithm

The PSO algorithm achieves a balance between exploration and exploitation , making it a robust and efficient method for solving complex optimization problems.

5.2 Surrogate-Assisted Optimization with Radial Basis Function Network (SuA-O: RBFN)

Surrogate-Assisted Optimization (Su-AO) approximates complex and computationally costly objective functions by using surrogate models. SAO is highly successful for large-scale optimization problems as it drastically reduces computational costs by either replacing or supporting the actual objective function.

In the context of Power Delivery Network (PDN) optimization, surrogate models enable the selection and placement of decoupling capacitors (decaps) with minimal computational overhead, speeding up the optimization process without sacrificing accuracy.

5.2.1 Radial Basis Function Network (RBFN)

The Radial Basis Function Network (RBFN) is a type of neural network used as a surrogate model in SAO. It consists of three layers:

- **Input Layer:** Receives the design specifications (e.g., electrical properties, decap locations).
- **Hidden Layer:** Computes non-linear mappings using radial basis functions (e.g., Gaussian, Multiquadric) as activation functions.
- **Output Layer:** Produces predictions by combining weighted outputs from the hidden layer.

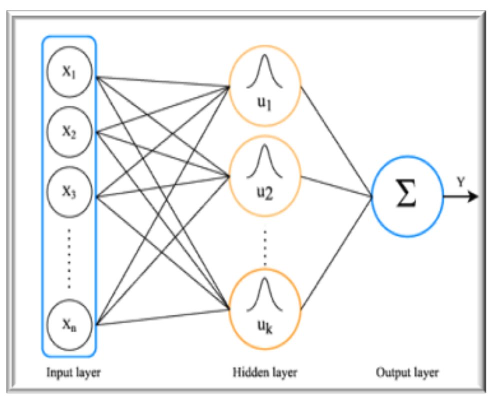


Figure 5.3: RBFN layers
[7]

Mathematical Representation: The output of the RBFN, $y(x)$, is calculated as:

$$y(x) = \sum_{i=1}^{N} w_i \cdot \phi(\|x - c_i\|) + b$$

Where:

- N: Number of neurons (radial basis functions) in the hidden layer.

- w_i: Weight of the connection between the i-th neuron and the output layer.
- $\phi(\|x - c_i\|)$: Radial basis function applied to the Euclidean distance between input x and the i-th center c_i.
  - Gaussian: $\phi(r) = e^{-\beta r^2}$
- c_i: Center of the i-th radial basis function.
- $\|x - c_i\|$: Euclidean distance between the input x and the center c_i.
- b: Bias term.

5.2.2 Framework for SuA-PSO with RBFN

The SuA-PSO (Surrogate-Assisted Particle Swarm Optimization) framework operates as follows:

(a) Train an RBFN model on a subset of the design space to approximate the objective function.

(b) During optimization:
   - Use the RBFN to evaluate particle fitness in PSO.
   - Evaluate only promising particles with the exact objective function based on RBFN predictions, reducing computational costs.

(c) Continuously refine the model with new data to improve accuracy.

5.2.3 The Algorithm

(a) **Initialization:**
   - Generate initial particles with random positions and velocities.
   - Train the RBFN surrogate model using a limited number of exact evaluations.

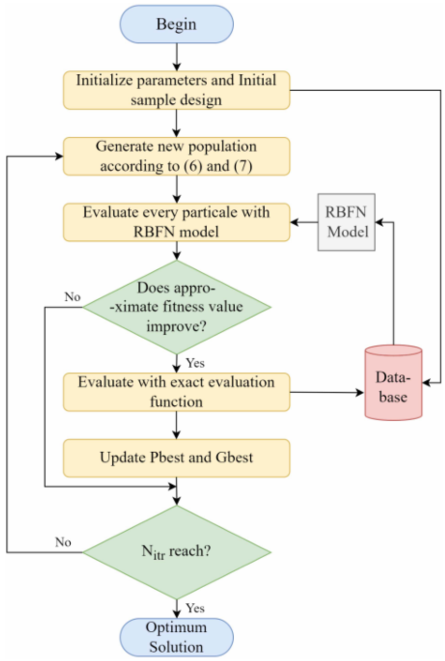
(b) **Evaluation of Surrogate-Based Fitness:**
   - Predict particle fitness using the RBFN.
   - Identify and evaluate promising particles using the exact objective function.

(c) **Update Process:**
   - Update the global and personal best positions (g_{best} and p_{best}).
   - Update particle velocities and positions.

(d) **Model Refinement:**
   - Retrain the RBFN using new data from exact evaluations.

(e) **Repeat:**
   - Continue iterating until stopping or convergence criteria are met.

(f) **Output:**
   - Provide the best solution available (g_{best}).



[8]

Figure 5.4: Flowchart of RBFN Algorithm

5.2.4 Benefits Compared to Standard PSO

[7]

- **Lower Computational Expense:** SuA-PSO significantly reduces CPU time by evaluating only promising solutions with the exact objective function.
- **Better Convergence:** The RBFN surrogate directs the swarm toward promising regions, speeding up convergence.
- **High Accuracy with Small Data:** RBFNs require fewer data points compared to other machine learning models, such as ANNs.
- **Scalability:** Suitable for complex combinatorial problems like PDN decap optimization.

This framework combines the computational efficiency of RBFNs with the exploration and exploitation capabilities of PSO, providing a robust solution for challenging optimization problems.

6 Simulation Study for PSO

In order to minimize impedance, this algorithm optimizes the positioning of decoupling capacitors, or decaps, in a Power Delivery Network (PDN) using Particle Swarm Optimization (PSO). Below is a breakdown of the key elements:

6.1 Setting up

- Particles are initialized with random starting positions and velocities.
- Limits for port assignments and capacitor indices are established.
- PSO parameters are set, such as the number of particles (N_p), number of ports, maximum iterations, inertia weight (ω), and cognitive/social coefficients (c_1, c_2).

Primary PSO Loop

- Runs for the maximum number of iterations while iterating across a specified number of ports (NOC).
- Updates particle position and velocity using PSO equations.
- Ensures constraints such as unique ports and valid port assignments are met.

Evaluation of the Objective Function

- Calculates the impedance for the specified capacitor and port configuration at each frequency.
- Captures the highest impedance (Z_{max}) across all frequencies.
- Checks whether Z_{max} satisfies the threshold (Z_t).

Updates on Best Value

- Updates the global best (g_{best}) and personal best (p_{best}) values for particles.
- Terminates the iteration early if a valid configuration is identified.

Visualization

- Plots the global best value for various NOCs against iterations.
- Creates a graph of impedance versus frequency for the optimal configuration.

6.2 Parameters Used for PSO in this Simulation

- **Dataset:**
  - Capacitors: `decaps.mat`
  - Y-matrix: `y.mat`
  - Frequency: `freq.mat`
- **PSO Parameters:**
  - Maximum frequency points (f_{max}): 1391
  - Number of particles (N_p): 50
  - Number of ports: 40
  - Maximum iterations: 50

– Total number of capacitors: 3348

– Impedance threshold (Z_t): 0.06 ohms

– Inertia weights (ω_i, ω_f): 0.9, 0.4

– Cognitive coefficient (c_1): 1.5

– Social coefficient (c_2): 1.5

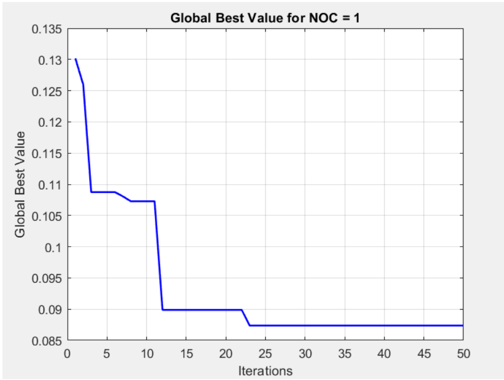
6.3 Result Plots

Global Best Value Against Iterations

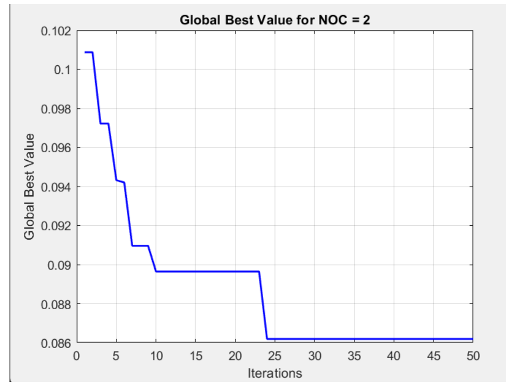
- Plots illustrating the convergence of the global best value for each NOC (1–6).
- Highlights the reduction in the global best value with iterations.
- Final values:
  - NOC = 1: $Z = 0.087365$
  - NOC = 2: $Z = 0.073147$
  - NOC = 3: $Z = 0.086186$
  - NOC = 4: $Z = 0.070801$
  - NOC = 5: $Z = 0.062367$
  - NOC = 6: $Z = 0.059266$

```
Time taken = 411.3006
Final Capacitor-Port Configuration:
Particle Index: 10
NOC: 6
Impedance: 0.059266
Capacitors:
  Columns 1 through 5

        1377         972         1521         1294         1138

  Column 6

        1159

Ports:
    16    10    15     7     8    11
```

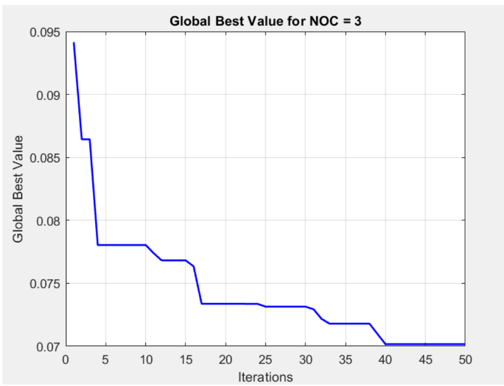
Figure 6.1: Simulation result showing no. of decaps,capictor id



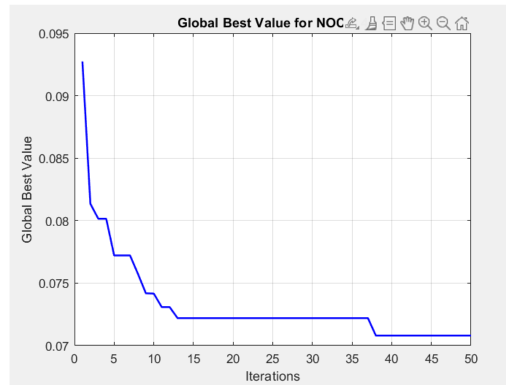
(a) NOC=1



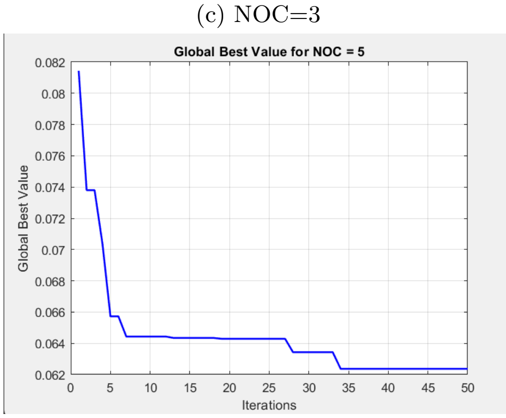
(b) NOC=2



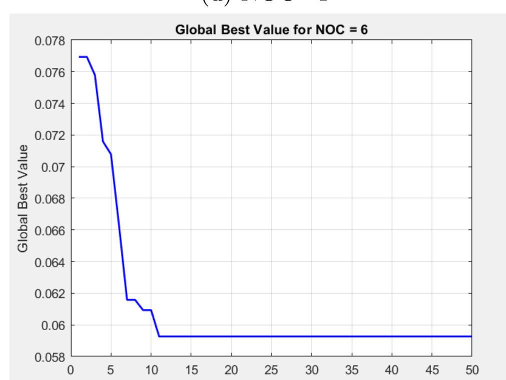
(c) NOC=3



(d) NOC=4



(e) NOC=5



(f) NOC=6

Figure 6.2: Performance plots for different NOCs (Number of Decoupling Capacitors)

Impedance vs. Frequency

- Displays the impedance curve for the optimal configuration.
- Shows minimal impedance at specific frequencies with a peak at higher frequencies.

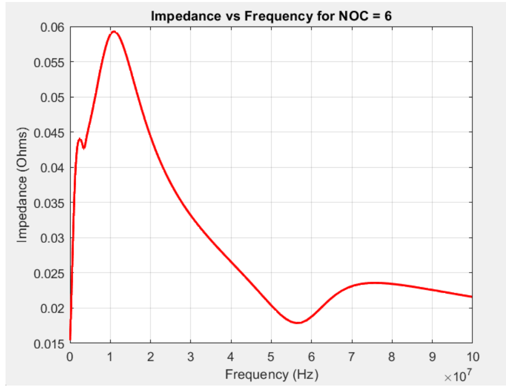


Figure 6.3: Impedance vs Frequency

6.4 Simulation Summary

- **Findings:** NOC = 6 produced the ideal arrangement.
- **Minimum Impedance:** $Z_{min} = 0.059226$
- **Indexes of Capacitors:** [1377,972,1521,1294,1138,1159]
- **Ports:** [16,10,15,7,8,11]
- **Time Taken:** $\sim 411.3\, seconds$

This simulation effectively illustrates the usefulness of PSO for PDN optimization, achieving excellent accuracy in impedance minimization and computational efficiency.

7 Simulation Study for SuA-PSO

In order to minimize impedance, this algorithm optimizes the positioning of decoupling capacitors (decaps) in a Power Delivery Network (PDN) using a surrogate-assisted Particle Swarm Optimization (PSO) approach. Specifically, a Radial Basis Function Network (RBFN) is employed as a surrogate model to approximate the impedance response, significantly reducing the computational cost. Below is a breakdown of the key elements:

7.1 Data Loading

- `decaps.mat`: Contains admittance profiles of decoupling capacitors. Each row is a capacitor and each column corresponds to a frequency point.
- `y.mat`: Stores the base PDN admittance matrix, frequency-dependent and port-wise structured.
- `freq.mat`: Frequency vector used for sweeping and impedance analysis.

7.2 Initialization

- Define PSO parameters: number of particles (N_p), maximum iterations (`maxiter`), number of ports, target impedance (Z_t), inertia weights (ω_i, ω_f), and acceleration coefficients (c_1, c_2).
- Define RBFN threshold `e_th` to decide whether to trust surrogate prediction or compute actual fitness.
- Initialize tracking arrays: `best_values_per_noc`, `overall_best_impedance`, `total_time`.

Outer Loop: Varying Capacitor Count (NOC)

- For each NOC from 1 to total number of ports:
  - Generate 100 random valid configurations with unique capacitor-port mappings.
  - Evaluate each using `calculate_fitness` over all frequencies.
  - Compute $Z_{eq} = inv(Y_{eq})$, extract impedance magnitude, and find Z_{max}.
  - Sort configurations by Z_{max}, pick top N_p as initial swarm.
  - Initialize RBFN with these configurations and corresponding fitness values.
  - Compute D_{max} and RBFN spread.

Inner PSO Loop

- Iterate for `maxiter` times:
  - Update inertia: $\omega = \omega_f + (\omega_i - \omega_f) \cdot \frac{\texttt{maxiter}-\texttt{iter}}{\texttt{maxiter}}$
  - Update velocity and position of each particle using PSO update rules.
  - Ensure valid and unique port assignments; round and clamp indices.
  - Predict fitness using RBFN:
    * If predicted value + `e_th` is better than personal best, skip actual evaluation.
    * Else compute real fitness, update personal/global bests, and expand RBFN database.
  - If $Z_{max} < Z_t$, terminate early.

Post-NOC Processing

- Store best impedance and configuration for each NOC.
- Plot impedance vs. frequency for optimal configuration.
- Log RBFN statistics: number of surrogate uses vs. real evaluations.
- Early terminate outer loop if target impedance met.

Final Output and Visualization

- Display: optimal NOC, best impedance, capacitor and port indices, total time, and evaluation counts.
- Plots:

- Impedance vs. frequency for all valid NOCs.
- Evolution of global best across iterations.

7.3 Parameters Used for SuA-PSO in this Simulation

- **Dataset:**
  - Capacitors: `decaps.mat`
  - Y-matrix: `y.mat`
  - Frequency: `freq.mat`
- **PSO Parameters:**
  - Maximum frequency points (f_{max}): 1391
  - Number of particles (N_p): 50
  - Number of ports: 40
  - Maximum iterations: 50
  - Total number of capacitors: 3348
  - Impedance threshold (Z_t): 0.06 ohms
  - Inertia weights (ω_i, ω_f): 0.9, 0.4
  - Cognitive coefficient (c_1): 1.5
  - Social coefficient (c_2): 1.5

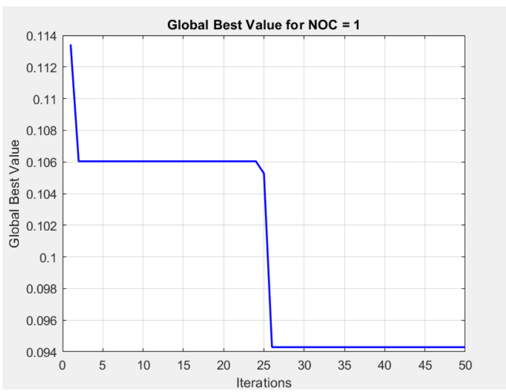
7.4 Result Plots for SuA-PSO approach

Global Best Value Against Iterations

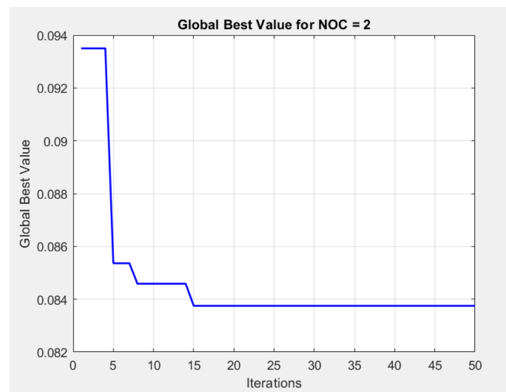
- Plots illustrating the convergence of the global best value for each NOC (1–6).
- Highlights the reduction in the global best value with iterations.
- Final values:
  - NOC = 1: $Z = 0.0943$
  - NOC = 2: $Z = 0.0838$
  - NOC = 3: $Z = 0.0683$
  - NOC = 4: $Z = 0.0632$
  - NOC = 5: $Z = 0.0593$

```
Optimum NOC: 5
Impedance for 5 de-caps: 0.059269
Final Capacitor-Port Configuration:
NOC: 5
Impedance: 0.059269
Capacitors:
         611        2252         932        1169        2274

Ports:
    16     10      8      6     11

Total Time taken for the entire optimization: 124.6841 seconds
```

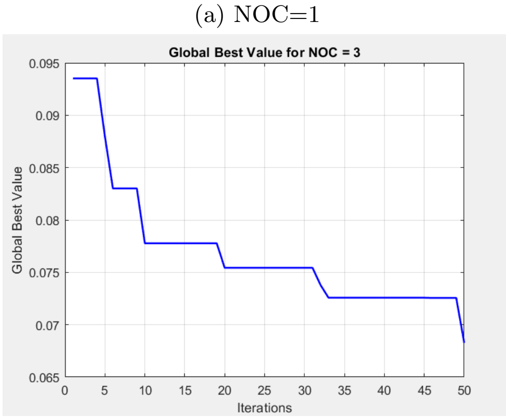
Figure 7.1: Simulation result showing no. of decaps,capictor id



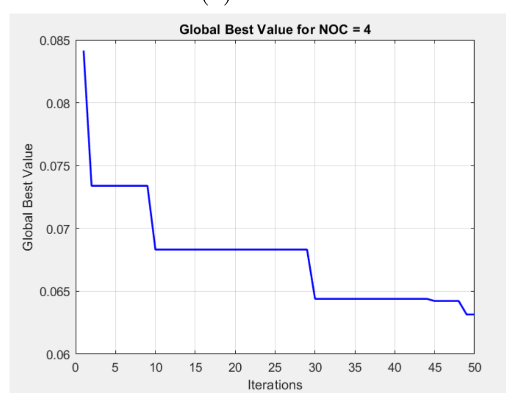
(a) NOC=1



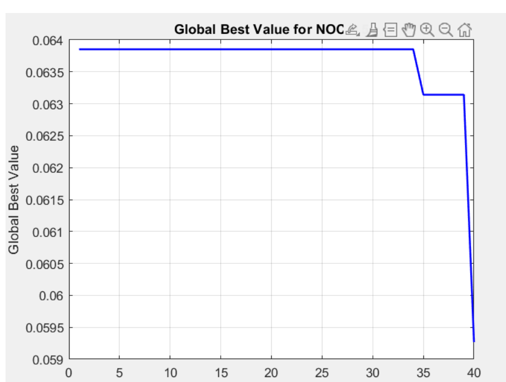
(b) NOC=2



(c) NOC=3



(d) NOC=4



(e) NOC=5

Figure 7.2: Performance plots for different NOCs (Number of Decoupling Capacitors)

Impedance vs. Frequency

- Displays the impedance curve for the optimal configuration.
- Shows minimal impedance at specific frequencies with a peak at higher frequencies for different NOC.

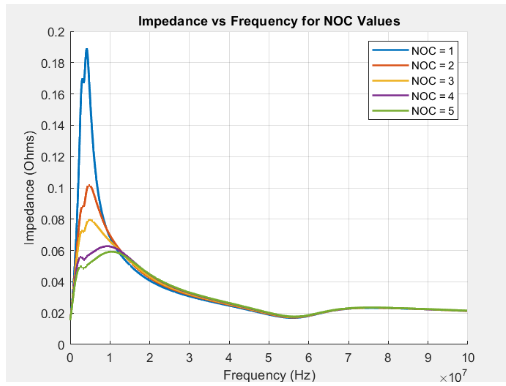


Figure 7.3: Impedance vs Frequency for each NOC

7.5 Simulation Summary

- **Findings:** NOC = 5 produced the ideal arrangement.
- **Minimum Impedance:** $Z_{min} = 0.059269$
- **Indexes of Capacitors:** [611,2252,932,1169,2274]
- **Ports:** [16,10,8,6,11]
- **Time Taken:** $\sim 124.68\, seconds$

This simulation effectively illustrates the usefulness of SuA-PSO for PDN optimization, achieving excellent accuracy in impedance minimization and computational efficiency.

8 Performance Comparison of Algorithm

8.1 Simulation Results for Ten Independent Runs with same dimension 3348x1391

Table 8.1: Performance Comparison between PSO and SuA-PSO (10 Runs)

PSO			SuA-PSO		
Z (mΩ)	**T (sec)**	**Nd (No. of decaps)**	**Z (mΩ)**	**T (sec)**	**Nd (No. of decaps)**
59.6	381	5	57.4	146	6
59.2	411	6	59.5	164	7
59.8	426	6	56.1	153	6
56.7	512	7	59.4	166	7
55.4	353	5	59.2	124	5
59.5	338	6	59.2	204	8
58.4	385	6	59.4	171	7
59.4	408	6	59.4	153	6
59.9	418	7	59.1	134	5
58.5	360	7	58.9	169	6

Table 8.2: Summary of Averaged Performance Metrics

Criterion	**PSO**	**SuA-PSO**
N_{avg} (Average no. of decaps)	6	6
N_{dmin} (Minimum no. of decaps)	5	5
T (Average Time in sec)	399.2	158.4
Gain in CPU Time	–	**60.3%**

Conclusion: SuA-PSO significantly reduced computation time compared to traditional PSO, achieving a **60.3% gain in CPU time**. Both algorithms achieved similar impedance performance and decap requirements. This demonstrates that *surrogate-assisted optimization* using RBFN can maintain solution quality while **greatly improving performance**.

9 Summary

This work addresses the problem of optimizing the placement of decoupling capacitors (decaps) in Power Delivery Networks (PDNs) to reduce impedance below a specified target threshold $Z_t = 60\, m\Omega$. While Particle Swarm Optimization (PSO) is effective for such non-linear optimization problems, it is computationally expensive due to repeated evaluations of the impedance over many iterations and particles.

To mitigate this, a Surrogate-Assisted PSO (SuA-PSO) approach is employed, where a Radial Basis Function Network (RBFN) is integrated as a surrogate model to approximate the impedance behavior. This significantly reduces the number of actual fitness evaluations, as the surrogate model is used to estimate fitness during optimization. Only when the RBFN predicts a sufficiently promising solution, the exact impedance is computed.

Key Observations from 10 Independent Runs

- **Decap Usage:** Both PSO and SuA-PSO achieved optimal configurations using an average of 6 decaps, with a minimum of 5.
- **Impedance Quality:** The achieved impedance values for both methods were comparable, e.g., $Z \approx 0.059\,\Omega$, meeting the target constraint.
- **Runtime Efficiency:**
  - PSO average runtime: **399.2 seconds**
  - SuA-PSO average runtime: **158.4 seconds**
  - Resulting in a **60.3% reduction in CPU time**
- **Best Configuration:** SuA-PSO produced optimal impedance using only 5 decaps and completed the entire optimization in under 125 seconds for a single run.

How SuA-PSO Achieves This Efficiency:

- Initial real evaluations are used to construct an RBFN surrogate model.
- During each iteration, candidate solutions are first evaluated using the RBFN.
- Real evaluations are only performed if the surrogate prediction exceeds a certain confidence threshold.
- The surrogate model is continuously updated with new real evaluations, improving accuracy.

Conclusion: The integration of RBFN with PSO in SuA-PSO leads to significant computational savings without degrading performance. By avoiding unnecessary evaluations, SuA-PSO accelerates convergence, making it highly effective for large-scale PDN decap optimization tasks. This hybrid approach strikes a balance between global exploration and local exploitation while maintaining high solution quality with reduced computational overhead.

References

[1] J. N. Tripathi, J. Mukherjee, P. R. Apte, N. K. Chhabra, R. K. Nagpal, and R. Malik, "Selection and placement of decoupling capacitors in high speed systems," *IEEE Electromagn. Compat. Mag.*, vol. 2, no. 4, pp. 72–78, Oct–Dec 2013.

[2] B. Magazine, "What is moore's law and how does it impact cryptography?" https://blockchainmagazine.com/what-is-moores-law-and-how-does-it-impact-cryptography/, accessed: Nov. 25, 2024.

[3] M. Saint-Laurent and M. Swaminathan, "Impact of power-supply noise on timing in high-frequency microprocessors," *IEEE Trans. Adv. Packag.*, vol. 27, no. 1, pp. 135–144, Feb 2004.

[4] J. N. Tripathi, J. Mukherjee, P. R. Apte, N. K. Chhabra, R. K. Nagpal, and R. Malik, "Selection and placement of decoupling capacitors in high speed systems," *IEEE Electromagnetic Compatibility Magazine*, vol. 2, no. 4, pp. 74–80, Oct 2013.

[5] H. H. Chen and D. D. Ling, "Power supply noise analysis methodology for deep-submicron vlsi chip design," in *Proc. 34th Annu. Des. Automat. Conf.*, 1997, pp. 638–643.

[6] S. Chun, M. Swaminathan, L. D. Smith, J. Srinivasan, Z. Jin, and M. K. Iyer, "Modeling of simultaneous switching noise in high speed systems," *IEEE Trans. Adv. Packag.*, vol. 24, no. 2, pp. 132–142, May 2001.

[7] H. Vaghasiya, A. Jain, and J. N. Tripathi, "A radial basis function network based surrogate-assisted swarm intelligence approach for fast optimization of power delivery networks," *IEEE Transactions on Signal and Power Integrity*, vol. 1, pp. 140–149, Oct 2022.

[8] B. Zhao *et al.*, "Decoupling capacitor power ground via layout analysis for multi-layered pcb pdns," *IEEE Electromagn. Compat. Mag.*, vol. 9, no. 3, pp. 84–94, Jul–Sep 2020.

[9] K.-B. Wu, A.-S. Liu, G.-H. Shiue, C.-M. Lin, and R.-B. Wu, "Optimization for the locations of decoupling capacitors in suppressing the ground bounce by genetic algorithm," *PIERS Online*, vol. 1, pp. 411–415, 2005.

[10] S. Hemaram and J. N. Tripathi, "Optimal design of a decoupling network using variants of particle swarm optimization algorithm," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2021, pp. 1–5.

[11] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw.*, 1995, pp. 1942–1948.

[12] Unknown, "Particle swarm optimization - figure from researchgate," https://www.researchgate.net/figure/Particle-swarm-optimization_fig3_354701029, accessed: Nov. 25, 2024.

[13] J. C. Bansal, P. K. Singh, M. Saraswat, A. Verma, S. S. Jadon, and A. Abraham, "Inertia weight strategies in particle swarm optimization," in *Proc. 3rd World Congr. Nature Biologically Inspired Comput.*, 2011, pp. 633–640.