Optimization of Decoupling Capacitors in VLSI Systems using Granularity Learning and Logistic Regression based PSO

Jai Narayan Tripathi[#], Dinesh Junjariya[#], Ramachandra Achar[*]
[#]Indian Institute of Technology, Jodhpur, Rajasthan, India
[*]Carleton University, Ottawa, Ontario Canada
{jai,junjariya.1}@iitj.ac.in, achar@doe.carleton.ca

***Abstract*— In order to reduce power supply fluctuations and to maintain a low Power Delivery Network (PDN) ratio in high-speed Very Large Scale Integration (VLSI) systems, decoupling capacitors are used in power delivery networks. In order to lower the cumulative impedance of the PDN below the target impedance, an Adaptive Granularity Learning (AGL) and Logistic Regression (LR) based Particle Swarm Optimization (PSO) is used for optimization of decoupling capacitors, in this work. The proposed approach provides results very efficiently compared to the state-of-the-art approaches. A maximum gain of 81% in terms of CPU time is achieved compared to the conventional PSO based approaches.**

***Keywords*— Power Integrity, Decoupling capacitor, Adaptive Granularity Learning Distributed Particle Swarm Optimization (AGLDPSO), Logistic Regression (LR).**

I. Introduction

Power Distribution Networks (PDNs) are responsible for supplying clean power within the silicon area in Very Large Scale Integration (VLSI) systems. A typical PDN consists of several components such as Voltage Regulator Module (VRM), power planes on board, bulk capacitors, nets in package, decoupling capacitors, etc. Right from the VRM to the silicon, there are several interconnects in the PDN. These interconnects having several limitations contribute to the nonidealities in the output response of the system. For example, due to inductance present in the PDN and the fluctuating current demand from the core circuitry inside silicon, voltage output of the PDN fluctuates. This kind of noise is termed as Simultaneous Switching Noise (SSN) and it prevents the PDN to deliver stable voltage to the load. The challenge in designing a PDN is to keep a stable voltage within specified voltage limits and it can be achieved by maintaining low impedance of the PDN.

In order to maintain power integrity in the system, the voltage ripples at the power supply of the load should be less than a specified voltage (ΔV_n). This voltage noise tolerance limit is defined based on the given maximum transient current requirement (I_{max_t}). Therefore, the impedance of the PDN needs to be maintained below some maximum allowable value across the frequency range, which is defined as the target impedance (Z_t) of the system. This is given by:

$$Z_t = \frac{\Delta V_n}{I_{max_t}} \tag{1}$$

In a defined frequency range, the probability of failure of system functionality depends on the PDN ratio at a rated performance. The PDN ratio is defined as the ratio of the maximum PDN impedance to the target impedance, and it should be less than or equal to unity. In order to maintain this, either some additional design steps or some more components in the PDN are required. The easiest and a cost effective way is to use additional capacitors in the PDN to lower down the PDN impedance. These are called as decoupling capacitors (decaps). Decaps are used on the package/board to achieve low Z_{pdn}, which helps in achieving PDN ratio less than unity, indicating a low probability of PDN performance failure.

The availability of multiple ports to place decaps and multiple capacitors to choose from, increases the possible port-decap combinations. Since testing for each port-decap combination can be very tedious when large number of capacitors are available, placement of decaps intuitively is not viable. The objective of the present work is to select set of ports and corresponding capacitors which can be placed efficiently, meeting the system requirement. Such large-scale optimization problems (LSOPs) have been addressed in the literature using computational intelligence based methods [1], [2]. Many studies including metaheuristic optimization and machine learning based techniques are available for optimal selection and placement of decoupling capacitors on corresponding ports [3], [4], [5]. In [4], ANN is used as an approximation of the objective function and it avoids real evaluation of objective function at all, which would increase the dependency of the optimization method on generating and training an efficient ANN model with very low error with respect to the real evaluation function. In [3], [5], [6], [7], reinforcement learning method is used, however the computational time taken by the method is not reported.

Machine learning (ML) is an effective way that helps to provide insights into any system with little knowledge of its working. As evolutionary computational (EC) algorithms process large data stored for search space, population and problem features during the iterative process, the ML techniques can come in handy in analyzing such data to improve the search performance. In this work, a Logistic Regression (LR) technique has been used for cluster control along with Adaptive Granularity Learning Distributed PSO (AGLDPSO), which includes locality-sensitive hashing (LSH) based clustering analysis [8].

979-8-3503-4764-7/23/$31.00 © 2023 IEEE

II. Problem statement

In this work, a practical system is considered for analysis. The system consists of a PDN supplying power to the core circuitry of a high-speed VLSI system. As a practice in industry, decoupling capacitors to be placed in the PDN are selected on the basis of their anti-resonance points and are placed as close to the core circuitry as possible [9]. This arrangement of capacitors helps to achieve the PDN impedance requirement eventually meeting the desired system performance. However, when several ports are available in the PDN for placing decaps, it becomes very challenging to select the decaps as well as the corresponding ports to place them. After placing the decaps, the equivalent self-impedance of the PDN, Z_{eq} can be computed as [10]:

$$Z_{eq} = (Z_{pdn}^{-1} + Z_{decap}^{-1})^{-1} \tag{2}$$

where Z-parameters matrix of the PDN is denoted by Z_{pdn}. Z_{pdn} has the dimension of $p \times p \times f$ where p and f represent the number of ports and frequency points, respectively. Z_{decap}, having same dimesion as Z_{pdn}, is a diagonal matrix having the self-impedance derived from the Z-parameters of the decaps. The impedance of decaps are indexed at the port number they are placed on as diagonal elements in Z_{decap}.

The aim of this optimization problem is to get the maximum self-impedance of the PDN below the target impedance for a given frequency range. As a result, this is a minimization problem and the objective function is the maximum value of the self-impedance over the frequency range, and is given as:

$$Z_{obj} = max(Z_{eq}(i, i, f)) \tag{3}$$

where f is a frequency point and i is the corresponding port where the self-impedance of the PDN is measured. The objective function in (3) is a function of port number and capacitors placed on those ports.

For the case study presented in this work, impedance of the PDN (Z_{pdn}) has a dimension of 21 $\times$ 21 $\times$ 1391, where the number 21 represents the number of ports and 1391 denotes the frequency points of study. The Z-parameters are derived from their respective S-parameters. This work has focuses on optimization of in-package decap since all the available ports are on the package. When no decoupling capacitor is used, the maximum self impedance value of the PDN is 361.2mΩ as shown in the Fig. 1.

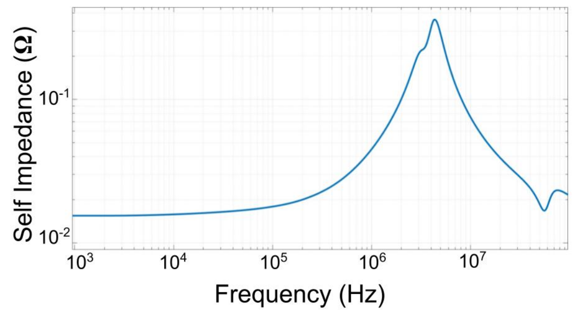

Fig. 1. Self-impedance of PDN (without decoupling capacitors)

III. Optimization of Decaps

For explaining the proposed approach, a brief background of relevant algorithms is given in this section.

A. PSO

Particle Swarm Optimization (PSO) is one of the popular metaheuristic algorithms used for many practical applications. This is inspired from the natural phenomenon such as birds flocking, fish schooling, etc. Swarm is also known as population in PSO and each sample/member of the population is called a particle [11]. A population is randomly defined and subsequently updated within the search space based on the velocity (V) of each of the particle. The velocity of a particle is dependent on its past performance and the performances of the other particles. The position (X) and velocity (V) of a particle can be given as:

$$V_{id}^{l+1} = \omega V_{id}^{l} + r_1 c_1 (X_{id}^{p} - X_{id}^{l}) + r_2 c_2 (X_{id}^{g} - X_{id}^{l}) \tag{4}$$

$$X_{id}^{l+1} = X_{id}^{l} + V_{id}^{l+1} \tag{5}$$

where the current iteration value is represented by l, D represents the number of dimensions ($d = 1, 2, ..., D$) in the problem; c_1 and c_2 are known as acceleration coefficients, while r_1 and r_2 are randomly generated numbers in the range of (0,1). $lbest$ is the local best position which represents the best position of the particle so far in terms of fitness value and is represented by X_{id}^{p}. The past position, which gives the best fitness value ($gbest$) among all the particles is represented as X_{id}^{g} and is referred as the current global best particle. The inertia weight (ω) is relevant to balance the global and local search abilities and it impacts the convergence of the optimization algorithm. In PSO, ω is varied from an initial smaller value to a greater value in the range of (0,1) during iteration as initially updated velocity is less dependent on the previous velocity, however, later in the process, it is logical to vary velocity in accordance to the previous velocity.

B. Adaptive Granularity Learning Distributed Particle Swarm Optimization (AGLDPSO) using Logistic Regression

To maintain the diversity of the population in the evolutionary process, distributed adaptive granularity learning is used with the particle swarm optimization method. The population of size N is divided into sub-population of size of M. After dividing the population into sub-populations, the worst particle (with the worst fitness value among the sub-population) is updated and the worst particle in a sub-population is co-evolved using PSO as :

$$V_{id}^{k+1} = \omega^{k} V_{id}^{k} + r_1 c_1 (X_{id}^{sp_j} - X_{id}^{k}) + r_2 c_2 (X_{id}^{g} - X_{id}^{k}) \tag{6}$$

$$X_{id}^{k+1} = X_{id}^{k} + V_{id}^{k+1} \tag{7}$$

where $X_{id}^{sp_j}$ is the particle with best fitness value in a sub-population and X_{id}^{g} is the global best particle in the whole population. The acceleration coefficients c_1 and c_2 are set to 1.5 and 0.5 respectively, to avoid premature convergence.

The clustering of the population is performed using locality-sensitive hashing (LSH). The basic idea behind the LSH is that two particles in the vicinity of each other have large probability to be adjacent in the new space by mapping. After the clustering analysis based on LSH, the granularity or the sub-population size M is adaptively varied on the basis of results from clustering analysis. Considering two cases of exploration and exploitation, when more number of particles are near to the worst particle in the population (compared to the global best particle), it represents exploration state. On the other hand, when more particles are near the global best particle (compared to the worst particle in the population), it represents exploitation state. The size M varies according to the state in which the population is distributed (according to the above two states). During exploration, M decreases to result in more number of sub-population and improves the population diversity, while during exploitation, M increases to accelerate the convergence of the population.

To vary the sub-population size M and also since there are two evolutionary states, a binary classifier is used. In this case, logistic regression is used to determine dichotomous outcome when multiple independent variables are present. The basis of LR is the sigmoid function which outputs decision into a probability [0,1] as:

$$s = \frac{1}{1+e^{-z}}; \tag{8}$$

where linear combination of input variables is denoted by z. LR is used to classify whether the algorithm is in the evolutionary state of exploration or exploitation.

C. Proposed Approach

In this work, it is proposed to use AGLDPSO to solve the classical optimization problem of decoupling capacitors. The flow of the proposed approach used for decap selection and placement can be well understood by Algorithm 1, where P and C are the ports and the corresponding capacitors selected for placement, to meet the system impedance requirement. N and M are the population and sub-population sizes, respectively; $maxFEs$ is the maximum number of fitness evaluation performed, $Y_{p\times p\times f}$ is the admittance matrix of the PDN. The new mapping for LSH in selecting decaps and ports were the anti-resonance points of the impedance profile of the particles. The sub-population size M is varied according to the result from LR. After the population is divided into sub-population, the worst particle in each sub-population is updated according to (6). The process is repeated till the the global best particle ($gbest$) of the population yields impedance lower than the target impedance.

IV. Results

The dimension of the dataset of decaps used in this case study is 1000×1391, where the number of decoupling capacitors present for placement is 1000 and the number of frequency points taken into account is 1391 (between 100 Hz to 100MHz). The dimension of cumulative impedance of the PDN matrix (Z_{pdn}) used is 21 × 21 × 1391, where 1391 denotes the frequency points taken into account and 21 are the number of ports available. Here, self impedance (Z_{11}) is measured at port-1 to keep the cumulative impedance minimum. The remaining 20 ports out of 21 ports are available for the placement of decaps. Z_t is kept at a value of 60mΩ to meet the system requirement.

Algorithm 1 AGLDPSO algorithm for decap optimization.

Input: $Y_{pdn} = Y_{p\times p\times f}, Z_t, N, c_1, c_2, \omega, maxFEs$
Output: Z_{obj}, C, P
$N_d = N_d + 1$;
Generate initial population X and velocity vector $V^{l=0}$;
Initialize sub-population size M;
$FEs = 0$;
while ($Z_{obj} > Z_j$) **do**
Update initial personal best population X^p
Check initial global best (minimum) Z_{obj}
  while ($FEs > maxFEs$) **do**
    $gbest$ particle is determined;
    Adaptively M is varied using Logistic Regression;
    Divide the population into N/M sub-population;
    for each sub-population **do**
      Determine the $pbest$ in the sub-population;
      Update the worst particle P_w;
      $FEs = FEs + 1$;
    end for
  end while
end while
Output: $Z_{obj} = max(Z_{eq}(1,1)$, where $(Z_{eq})_f = (Y_{pdn}^{-1} + Y_{decap}^{-1})_f^{-1}$, $\forall \in$ [0, f_{max}] , $X^g = [P\ C]$

For comparison with the state-of-the-art, conventional PSO is used along with the Matrix-based PSO [12]. To compare the proposed method with other machine learning based metaheuristic algorithms, a radial basis function based surrogate assisted PSO (SuA-PSO) is also considered, which uses neural network as a surrogate model for fitness value evaluations [13]. The population size of 50 has been set for PSO and MPSO, for the proposed method population size of 500 has been considered, while for SuA-PSO, the population size is set to be 20. For the proposed approach, maximum number of fitness evaluations were set at 10000. For PSO, MPSO and SuA-PSO maximum number of iterations has been set to 50. All the algorithms were executed on a system with 8 GB of RAM and Intel i5 8^{th} Gen 2.4 GHz on MATLAB R2019b.

For all four algorithms, 10 independent runs were considered for performance comparison and their summary is presented in Table-1. In Table-1, N_{avg} and N_{min} denotes the average and minimum number of decoupling capacitors used, respectively, for meeting system performance. Here, Z is the impedance value of PDN corresponding to N_{min} and

T (in sec) is the average computational run time for the algorithm to converge. Using the proposed approach, a gain in CPU run time of 81.2% is observed compared to the PSO, 79.4% compared to the MPSO, and 52% compared to the surrogate-assisted PSO approach is observed.

Table 1. Performance summary of algorithms

Parameter	Optimization Algorithms			
	PSO [11]	MPSO [12]	SuA-PSO [13]	Proposed approach
N_{avg}	7	6	8	6
N_{min}	5	5	6	4
Z	59.3	54.2	58.2	58.3
T	731	668	286	137
Gain in CPU Time	**81.2%**	**79.4%**	**52%**	-

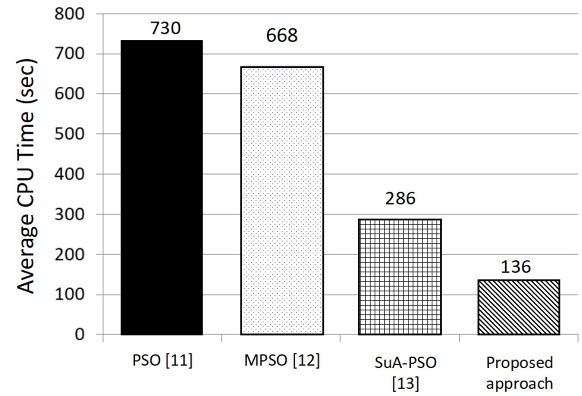

Fig. 3. Comparison of average computational time

The optimum self impedance of the PDN obtained by PSO, MPSO and the proposed AGLDPSO based approaches using minimum number of decoupling capacitors is shown in Fig. 2. In Fig. 2, it can be observed that in the proposed approach, diversity of the population while optimum search is maintained, resulting in convergence at minimum number of decaps i.e. 4 to meet system requirement, while for PSO and MPSO, it is 5. The average and minimum number of decaps required to meet the target impedance for conventional PSO are 7 and 5. For MPSO, average and minimum number of decaps required to meet the target impedance are 6 and 5, while for SuA-PSO they are 8 and 6. However in the proposed approach, average and minimum decaps required are the minimum among all the other three approaches, i.e. 6 and 4, respectively. Thus, the proposed technique of AGLDPSO has resulted in significant reduction in computational time as summarised pictorially by Fig. 3.

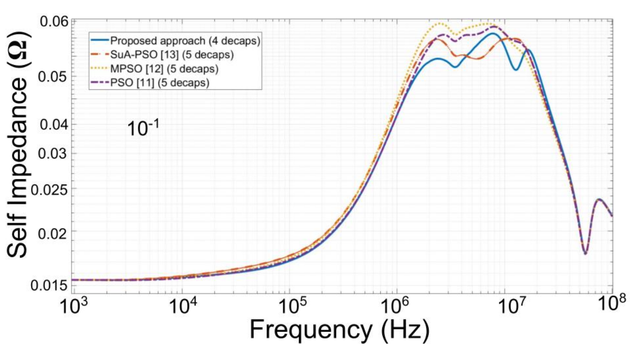

Fig. 2. Optimum Impedance of the PDN

V. Conclusion

In this work, a fast and efficient method for decoupling capacitor placement and selection in a PDN is discussed using granularity learning and logistic regression. In this study, a practical power distribution network is taken into account, and the target impedance of the system is achieved using the least possible number of decoupling capacitors. The proposed method uses adaptive sub-population sizing with the help of logistic regression to improve computation time for large scale optimization problem of decoupling capacitors placement compared to the conventional PSO techniques and neural network based surrogate assisted PSO. Proposed approach outperforms other traditional metaheuristic algorithms in terms of fast convergence and reduced computation time efficiency.

Acknowledgment

This work is supported by the SRG Scheme of Science and Engineering Research Board (SERB), Govt. of India. Project No.: SRG/2021/002239.

References

[1] R. Kruse *et al.*, *Computational intelligence*. Springer, 2011.
[2] M. Gendreau *et al.*, "Metaheuristics in combinatorial optimization," *Annals of Operations Research*, vol. 140, no. 1, pp. 189–213, 2005.
[3] R. Cecchetti *et al.*, "Effective pcb decoupling optimization by combining an iterative genetic algorithm and machine learning," *Electronics*, vol. 9, no. 8, p. 1243, 2020.
[4] S. Han *et al.*, "Reinforcement learning for the optimization of decoupling capacitors in power delivery networks," in *2021 IEEE International Joint EMC/SI/PI and EMC Europe Symposium*. IEEE, 2021, pp. 544–548.
[5] H. Park *et al.*, "Deep reinforcement learning-based optimal decoupling capacitor design method for silicon interposer-based 2.5-d/3-d ics," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 10, no. 3, pp. 467–478, 2020.
[6] E.-G. Talbi, "Machine learning into metaheuristics: A survey and taxonomy," *ACM Computing Surveys (CSUR)*, vol. 54, no. 6, pp. 1–32, 2021.
[7] J. A. Easum *et al.*, "Multi-objective surrogate-assisted optimization applied to patch antenna design," in *2017 IEEE international symposium on antennas and propagation & USNC/URSI national radio science meeting*. IEEE, 2017, pp. 339–340.
[8] Z.-J. Wang *et al.*, "Adaptive granularity learning distributed particle swarm optimization for large-scale optimization," *IEEE transactions on cybernetics*, vol. 51, no. 3, pp. 1175–1188, 2020.
[9] J. N. Tripathi *et al.*, "Selection and placement of decoupling capacitors in high speed systems," *IEEE Electromagnetic Compatibility Magazine*, vol. 2, no. 4, pp. 72–78, 2013.
[10] E. Kulali *et al.*, "Chip power model-a new methodology for system power integrity analysis and design," in *2007 IEEE Electrical Performance of Electronic Packaging*. IEEE, 2007, pp. 259–262.
[11] S. Hemaram *et al.*, "Metaheuristic optimization of decoupling capacitors in a power delivery network," in *2021 IEEE International Joint EMC/SI/PI and EMC Europe Symposium*. IEEE, 2021, pp. 554–558.
[12] A. Jain *et al.*, "Efficient selection and placement of in-package decoupling capacitors using matrix-based evolutionary computation," *IEEE Open Journal of Nanotechnology*, vol. 2, pp. 191–200, 2021.
[13] H. Vaghasiya *et al.*, "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, 2022.