IEEE Open Journal of Nanotechnology

Received 5 November 2021; revised 23 November 2021; accepted 25 November 2021. Date of publication 7 December 2021; date of current version 22 December 2021. The review of this paper was arranged by Associate Editor Brajesh K. Kaushik.

Digital Object Identifier 10.1109/OJNANO.2021.3133213

Efficient Selection and Placement of In-Package Decoupling Capacitors Using Matrix-Based Evolutionary Computation

AKASH JAIN, HEMAN VAGHASIYA, AND JAI NARAYAN TRIPATHI (Senior Member, IEEE)

Department of Electrical Engineering, Indian Institute of Technology Jodhpur, Jodhpur 342037, India

CORRESPONDING AUTHOR: JAI NARAYAN TRIPATHI (e-mail: jai@iitj.ac.in)

(Akash Jain and Heman Vaghasiya contributed equally to this work.)

ABSTRACT In the era of advanced nanotechnology where billions of transistors are fabricated in a single chip, high-speed operations are challenging due to packaging related issues. In High-Speed Very Large Scale Integration (VLSI) systems, decoupling capacitors are essentially used in power delivery networks to reduce power supply noise and to maintain a low impedance of the power delivery networks. In this paper, the cumulative impedance of a power delivery network is reduced below the target impedance by using state-of-the-art metaheuristic algorithms to choose and place decoupling capacitors optimally. A Matrix-based Evolutionary Computing (MEC) approach is used for efficient usage of metaheuristic algorithms. Two case studies are presented on a practical system to demonstrate the proposed approach. A comparative analysis of the performance of state-of-the-art metaheuristics is presented with the insights of practical implementation. The consistency of results in both the case studies confirms the validity of the proposed appraoch.

INDEX TERMS Power delivery networks, power integrity, decoupling capacitors, metaheuristic algorithms, matrix-based evolutionary computation (MEC).

I. INTRODUCTION

With the advent of nanotechnology followed by the vision presented in the famous lecture of R. Feynman,[1] a significant development due to miniaturization in all the fields of technology started. A drastic advancement of nanotechnology over the last couple of decades has enabled multifunctional electronic devices for end-market users by very large scale integration (VLSI). This has been possible due to the transistors having dimensions in nanometer range which facilitate very fast rise/fall times and/or operating frequencies. The sizes are still shrinking and so far, Moore's law has predicted this trend very well. However, in the current state-of-the-art high-speed designs, the impact of *nano* dimensions is much more prominent than any other aspect in the VLSI systems [2]. Because of the nanoscale dimensions of switching devices or transistors, the design margins have become very narrow. On top of that, the bandwidth limitation of the interconnects is making it worse.

Today, when the 5 *nm* VLSI technology based end-products are already available in the consumer market, the nanotechnology and VLSI communities should not be seen as separate entities. There have already been reported extensive works on interconnects in the nanotechnology community, which are majorly focused on Graphene and Carbon based nanotubes [3]–[6]. In the literature, it is noteworthy that the relevance of nanotechnology at the circuit level can also be seen even from the lower level of abstraction such as from the atomic level [7]. Vertical Nanowire FETs (V-NWFET) are also being developed as an alternatives targeted for high-speed applications in 5 *nm* and beyond technology nodes [8].

In addition to this, in context of nanotechnology it should be noted that, as the advanced memory cells such as Resistive Random Access Memory (RRAM), Phase Change Memory (PCM), etc. are being developed using nanotechnology, the packaging effects should be analysed essentially to ensure the robustness of the system [9]. The interconnects within the Nano Integrated Circuits face several SI issues such as crosstalk, reflection, EMI effects, surface roughness,

[1]There's Plenty of Room at the Bottom! [1].

etc. [10]–[13]. Therefore, there is a strong need to study the effects of interconnect on system performance in terms of SI and PI. The bandlimited interconnects which are present both at on-chip and off-chip are responsible for Signal Integrity (SI) and Power Integrity (PI) issues in the system. SI corresponds to the quality of signal propagated through interconnects from one point to another point in the system while PI corresponds to the quality of power distribution within the system. The continuous scaling down of transistors allows accommodating billions of transistors in present integrated circuits (ICs), leading to a huge increase in current and a correspondingly a significant decrease in noise margins. The simultaneous switching of millions of transistors together makes prominent fluctuations in power supply.

Power delivery networks (PDNs) are designed to provide low-noise DC power supply to the active components of the system. As the supply voltage drops, the variance in power supply produced by current transients becomes more prominent. The dynamic change of the supply voltage is a major contributor to supply noise. Maintaining the PI is an important step in system design since supply noise can affect the speed of the ICs and the performance of the PDN at various frequency ranges [2], [14]–[17].

The voltage noise and worst-case transient current together provide a limit for the maximum permissible PDN impedance, ensuring that the supply noise never exceeds the specification. In PDN design, the maximum allowable PDN impedance is termed as target impedance(Z_T) which is given by:

$$Z_T = \frac{\Delta V}{I_{max-t}} \tag{1}$$

where ΔV is the maximum specified voltage rail noise to meet performance requirements and I_{max-t} is the worst-case transient current under any possible operation. PDN ratio, a ratio of PDN impedance to the target impedance, is a rough estimate of the probability of failure of circuits to function at rated performance. Low risk of PDN-related failure is suggested by a ratio of less than one. As this ratio rises, the risk of failure of PDN rises with it. In general, achieving low PDN impedance and lower PDN ratio is expensive as it requires the use of more components, more layers in the package/board, or the use of more expensive materials. Placing decoupling capacitors (commonly called as 'decaps') on the board and/or on packages is a simple and cost-effective option to achieve low PDN impedance. Identification of anti-resonance frequencies in the PDN impedance profile guides the intuitive selection of decoupling capacitors.

The intuitive selection of decoupling capacitors is guided by the identification of anti-resonance frequencies in the PDN impedance profile. When there are numerous ports on the board and multiple capacitors accessible, the intuition-based approach is not effective. Computational Intelligence (CI) based techniques have been proved to be very useful to address such large-scale combinatorial problems. There are several studies that use various optimization approaches to optimise the selection and placement of decoupling capacitors [18]–[30]. However, the primary drawback of the metaheursitics based approach is its time complexity. The time complexity of optimization problem can be reduced by using recently introduced Matrix-Based Evolutionary Computation approach [31].

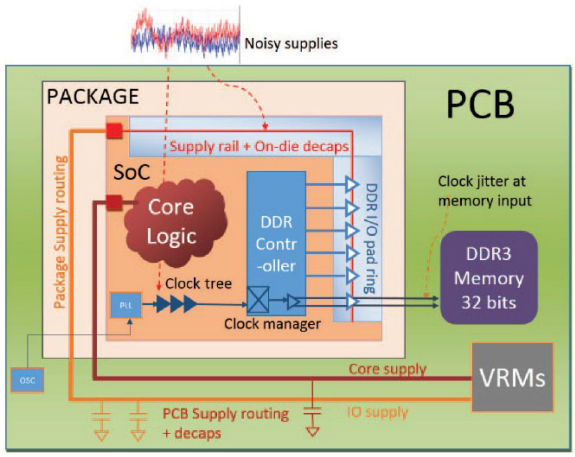

FIGURE 1. System considered for Analysis [32].

This paper presents two case studies demonstrating optimal selection and placement of in-package decoupling capacitors using various swarm intelligence-based metaheuristic optimization techniques to minimize the self impedance of the PDN below the Z_T to maintain a low PDN ratio. The Performance comparison of several algorithms is also presented in this paper. The optimization problem is formulated using s-parameter data from PDN and decoupling capacitors for this work. The rest of the paper is organised as following. Section II presents description of system used for the problem statement. Section III presents the overview of metaheuristic algorithms used in the paper. In Section IV, optimization problem is discussed while Section V shows results and comparative analysis of the algorithms. Section VI concludes the paper.

II. SYSTEM DESCRIPTION

In this paper, for both the case studies, a practical system is used. The block diagram of the system is shown in Fig. 1. In this system, a DDR3 memory is used for which a memory controller IC controls the READ/WRITE operations. In the system, there are two power supplies: one for the digital core and another one for the I/O circuits. This controller IC has many internal blocks which work based on the instructions obtained from a clock tree. In this paper, the PDN of the core circuit is taken into account for both the case studies. As shown in Fig. 1, the clock gets power supply from this PDN. If there is noise in the PDN, it will eventually affect the timing of the clock tree. The jitter introduced due to the PDN noise can cause the setup and hold time violations in the system. This power supply is very critical as the fluctuations in this supply

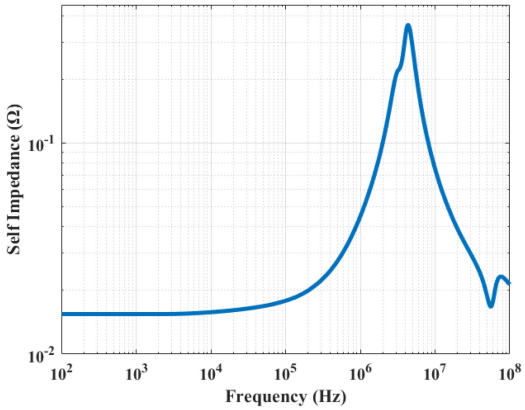

FIGURE 2. Self-impedance of PDN without Decoupling capacitors.

will cause timing variations in the rise/fall times which can eventually affect the READ/WRITE operations and may even fail the system.

As mentioned earlier, it's a common practice to choose capacitors intuitively based on the antiresonance points in the PDN impedance profile, choosing the values of capacitors accordingly and then placing them as near as possible to the core circuit [33]. In the PDNs where there are multiple ports available for placing decaps and there is a choice of plenty of capacitors available to choose from, it becomes very difficult to place decoupling capacitors efficiently. The objective of this work is to minimize the number of decoupling capacitors while maintaining the impedance of the PDN lesser than the target impedance.

Fig. 2 illustrates the self-impedance of the PDN considered, measured at the pad of the IC. It is the cumulative impedance of all the elements of PDN such as Voltage Regulator Module (VRM) and interconnects on package, PCB, and chip. A commercial 3D solver is used to extract s-parameters for the PCB and package models and a Chip Power Model (CPM) [34] is employed for the on-chip PDN model. The dimensions of Z_{PDN} (Z-parameters are computed from the S-parameters) used for the analysis is $21 \times 21 \times 1391$, where 21 indicates the number of ports and 1391 denotes the number of frequency points. All the ports available are inside the package so this paper deals with in-package decap optimization. The maximum value of self impedance of PDN is $361.2m\Omega$ when no decoupling capacitor is present, as shown in Fig. 2.

In this paper, two datasets of decoupling capacitors are used for two case studies. The dimensions of decaps for first dataset (Case-study 1) is 8×1391 and for second dataset (Case-study 2) is 3348×1391. The first dataset consists of 8 capacitors. Table 1 lists the capacitors used for Case-study 1 with their indices and their other details. The dataset for Case-study 2 consists of 3348 capacitors manufactured by AVX. The impedance Z_{11} is measured at port-1 (which is the port the closest to the IC) to minimize the equivalent self impedance of the PDN. Thus there are

TABLE 1. Decaps for Case-Study 1

C	Capacitor Model	Manufacturer
1	GCM155R72A472KA37	Murata
2	GRT155R61E105KE01	Murata
3	CL05B104KO5NNN	Samsung
4	GRT155R61E105KE01	Murata
5	CL05B103KO5NNN	Samsung
6	GRT1555C1H100JA02	Murata
7	C1005X5R0J475M050BC	TDK
8	GRM1555C1H330JA01	Murata

20 ports accessible for decaps. Z_T is $60m\,\Omega$ to meet system requirements.

The equivalent cumulative impedance matrix, Z_{eq}, can be computed as when the decoupling capacitors are placed at the corresponding port [35]:

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

where Z_{pdn} represents the Z-parameter matrix of PDN, Z_{Decap} represents the Z-parameter of the decaps which is a diagonal matrix. Z_{pdn} has a decaps impedance at diagonal elements that corresponds to the port number where they are placed. The following is an alternative relationship for equation (3) [33]:

$$Z_{eq} = (Y_{pdn} + Y_{Decap})^{-1} \tag{3}$$

The objective of the optimization problem is to reduce Z_{11} to less than Z_T. Therefore, the maximum value of a self impedance of the PDN is the objective function for this analysis, which is defined as follows:

$$Z_{obj} = max(Z_{eq}(1, 1)). \tag{4}$$

Thus, it is a minimization problem with (4) as an objective function. There might be constraints also based on the application e.g. using as small a number of decaps as possible, etc.

In the next section, a brief overview of various metaheuristic algorithms used in this paper is presented.

III. METAHEURISTIC OPTIMIZATION

Metaheuristic algorithms employ stochastic optimization and don't need gradient computations [36]. As a result, these algorithms do not require the optimization problem to be differentiable, instead, the solution is based on the collection of random variables created. When compared to other optimization methods, these techniques find suitable solutions with less computing. Swarm Intelligence metaheuristics [37] are Evolutionary Computation (EC) algorithms that determine the optimal solution based on the collective behavior of decentralized, self-organized candidates in a population or swarm. Swarm-based metaheuristics use a common framework to find the global optimum by a process based on a population. The process includes initialization, objective function evaluation, and new population reproduction. Therefore, swarm-based metaheuristics are population-based and iteration-based algorithms with more outstanding global search capabilities than single-solution search algorithms like Simulated Annealing

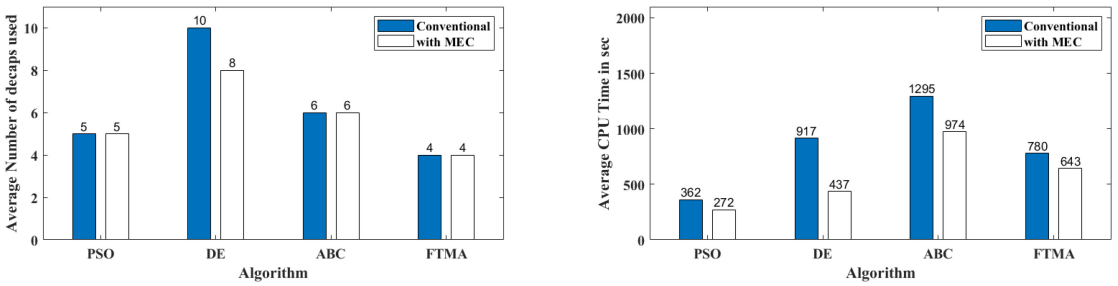

FIGURE 3. Average number of decaps and Time comparison for Case-Study 1.

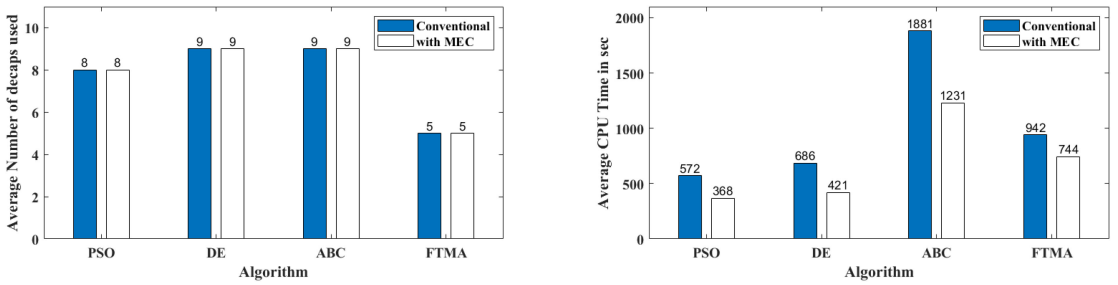

FIGURE 4. Average number of decaps and Time comparison for Case-Study 2.

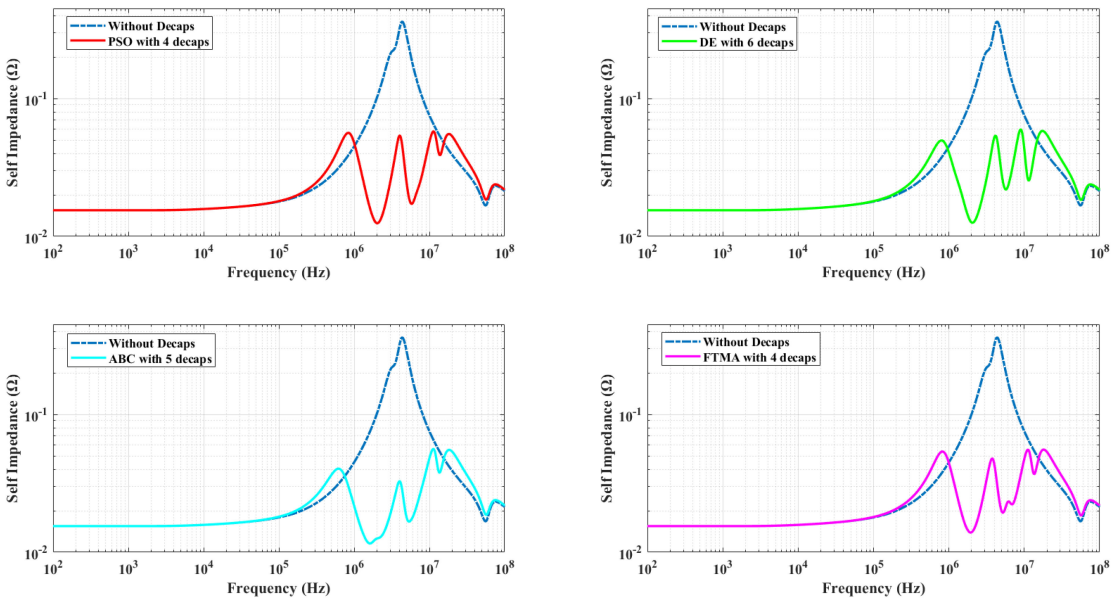

FIGURE 5. Optimal Impedance of PDN for Case-Study 1.

(SA). Particle Swarm Optimization (PSO), Differential Evolution Algorithm (DE), Artificial Bee Colony (ABC), and Fine-Tuning Meta-Heuristic Algorithm (FTMA) are the state-of-the-art Swarm Intelligence metaheuristics used in this work.

A. PARTICLE SWARM OPTIMIZATION

Kennedy and Eberhart introduced Particle Swarm Optimization (PSO) [38], [39] in 1995 as a nature-inspired optimization approach based on the swarm behavior of birds' flocks or fish schools. PSO is one of the most widely used swarm-intelligence-based algorithms due to its simplicity and flexibility. Particle Swarm Optimization with the Linear Decreasing Inertia Weight (LDIW) [40] is used in this paper to address the optimization problem. Swarms, or particles in PSO, are created at random, with each particle representing a solution to the objective problem inside the given search

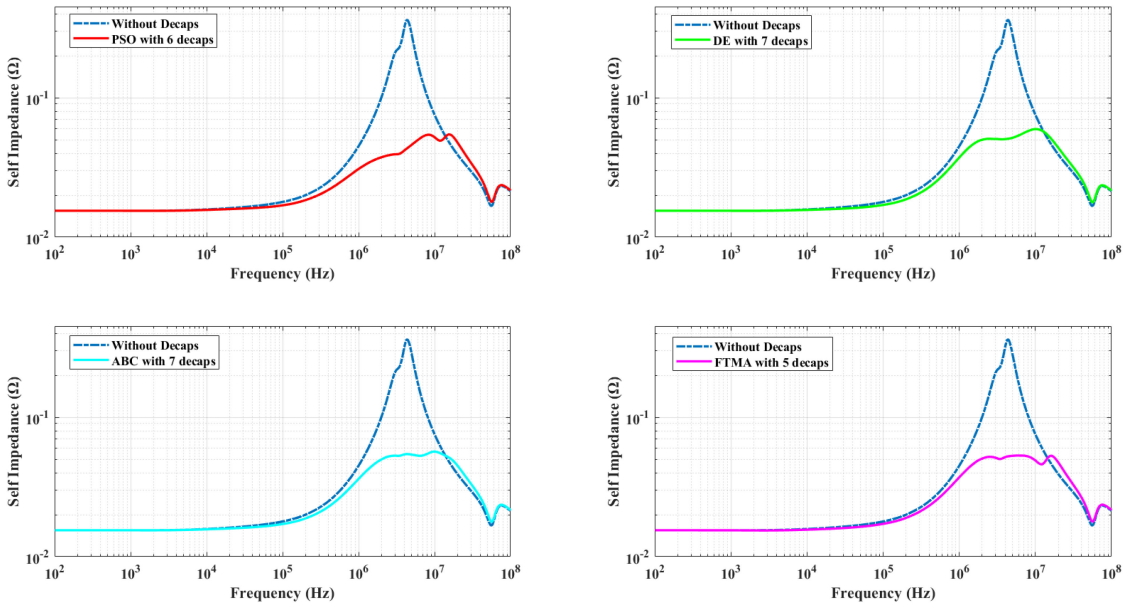

FIGURE 6. Optimal Impedance of PDN for Case-Study 2.

space. In the PSO algorithm, particle movements are controlled by their velocities (V) and locations (X), which are as follows:

$$V_{kd}^{n+1} = w\,V_{kd}^{n} + r_1 c_1 (X_{kd}^{p} - X_{kd}^{n}) + r_2 c_2 (X_{kd}^{g} - X_{kd}^{n}) \quad (5)$$

$$X_{kd}^{n+1} = X_{kd}^{n} + V_{kd}^{n+1} \quad (6)$$

where N is the number of particles in the population, D denotes the number of dimensions i.e. $d = 1, 2...D$; n denotes the current iteration, X_{kd}^{p} each particle's individual or personal best, and X_{kd}^{g} the current global best. c_1 and c_2 are acceleration coefficients, while r_1 and r_2 are random numbers in the [0,1] range. Here, W represents the inertia weight. For LDIW-PSO inertia weight is given as follows:

$$w = w_f + (w_i - w_f) \times \frac{MaxIt - n}{MaxIt} \quad (7)$$

where w_i and w_f denote the initial and final inertia weights, respectively, and $MaxIt$ denotes maximum number of iterations.

B. DIFFERENTIAL EVOLUTION ALGORITHM

Storn and Price introduced the Differential Evolution Algorithm (DE) in 1997 [41], [42], which leverages prior candidates' solutions to generate new ones. The DE algorithm is a self-organizing stochastic search method that does not employ derivative information. As a result, it's a population-based, derivative-free approach. Due to its use of crossover and mutation, DE is similar to pattern search and genetic algorithms. In DE, mutation of population (V) and crossover of population (U) is given as follows:

$$V_k^{n+1} = X_{r1}^{n} + F(X_{r2}^{n} - X_{r3}^{n}) \quad (8)$$

$$U_{kd}^{n+1} = \begin{cases} V_{kd}^{n+1} & \text{if } (rand \geq pCR) \text{ or } j = randi(k) \\ X_{kd}^{n} & \text{if } (rand > pCR) \text{ and } j \neq randi(k) \end{cases} \quad (9)$$

where $d = 1, 2, \ldots, D$ for n^{th} particle, and $r1$, $r2$, and $r3$ are random integers in population size N and $\neq k$. Here, F and pCR represent the amplification and crossover constants, respectively; $rand$ is random number in [0,1] and $randi(k)$ is a randomly chosen index from 1 to D. For selection of new population, the greedy criterion is used.

$$X_k^{n+1} = \begin{cases} U_k^{n+1} & \text{if } Obj(U_k^{n+1}) < Obj(X_k^{n}) \\ X_k^{n} & \text{otherwise} \end{cases} \quad (10)$$

C. ARTIFICIAL BEE COLONY

Artificial Bee Colony (ABC), introduced by Karaboga and Basturk in 2007 [43], [44], is another nature-inspired optimization approach based on the intelligent foraging behavior of a honey bee swarm. In the ABC algorithm, the employed artificial bees make up half of the colony, while the observers make up the other half. There is only one hired bee for each food source. In other words, the number of employed bees is equivalent to the amount of food sources in the immediate vicinity of the hive. When the employed and onlooker bees have exhausted their food supply, the employed bee becomes a scout. Normally, the number of onlooker bees (N_O) is taken equal to the number of employed bees i.e. population size (N). The ABC use the following expression to generate a candidate food position from an existing one:

$$X_{kd}^{n+1} = X_{kd}^{n} + \phi_{kd}(X_{kd}^{n} - X_{rd}^{n}) \quad (11)$$

where $r \in 1, 2, .., N$ is randomly chosen index and $r \neq i$, and ϕ_{kd} is random number $\in [-1, 1]$. After all employed

bees have completed their search, they communicate information about their food sources with onlooker bees. An onlooker bee assesses the nectar data collected from all employed bees and selects a food source with a probability related to its nectar quantity. This probabilistic selection is actually a roulette wheel selection process, as seen in the following equation:

$$P_k = \frac{Obj_k}{\sum Obj_j} \tag{12}$$

If a position cannot be improved further after a certain number of cycles (known as the Abandonment Counter Limit L), the food source is considered abandoned and then the scout bee discovers a new food source.

D. FINE-TUNING META-HEURISTIC ALGORITHM

Fine-Tuning Meta-Heuristic Algorithm (FTMA) [45] is a solution update algorithm introduced by Allawi, Ibraheem, and Humaidi in 2019. It utilizes exploration, exploitation, randomization, and selection sequentially for solution updating. The formula for exploration is as follows:

$$Y_{kd} = X^n_{kd} + rand \times (X^n_{kd} - X^n_{kd}) \tag{13}$$

where $r \in 1, 2, .., N$ is a randomly chosen index and $r \neq k$. The temporary fitness g is then calculated using the value of the objective function for the temporary solution Y_k. Now, exploration equation is as follows:

$$\begin{aligned} &\text{if } g > Obj(X^n_k) \text{ and } p > rand, \\ &Y_{kd} = X^n_{kd} + rand \times (X^n_{kd} - X^n_{kd}) \end{aligned} \tag{14}$$

where p is the exploitation probability; *rand* is a random number $\in [0, 1]$, and X^n_b is the current global best solution. So, new g is calculated using new Y_k. Now, if $g > Obj(X^n_k)$ and $r > rand$, new randomize Y_k is generated and g is evaluated, where p is the randomization probability. The final step i.e. selection step is as follows:

$$X^{n+1}_k = \begin{cases} Y_k & \text{if } g < Obj(X^n_k) \\ X^n_k & \text{otherwise} \end{cases} \tag{15}$$

$$X^{n+1}_b = \begin{cases} Y_k & \text{if } g < Obj(X^n_k) \\ X^n_b & \text{otherwise} \end{cases} \tag{16}$$

IV. OPTIMIZATION

In this section, a general solution of the optimization problem introduced in Section II is discussed. Later, a computationally superior approach is used to improve the general solution.

A. METAHEURISTICS BASED SOLUTION

The objective function for the optimization problem (equation (5)) is a function of two variables: the port number and the index of the decoupling capacitor corresponding to that port; this means that each capacitor has two decision variables, D. For N number of population and $MaxIt$ number of maximum iterations, the output of each algorithm would be the optimum value of self impedance of the PDN (Z_{obj}), the number of decaps (n_D), and decoupling capacitors with their corresponding indices (C) and port numbers (P). Algorithm-1 summarises pseudo code for the decoupling capacitor optimization methodology in a PDN, where p represents the number of ports and f is the number of frequency points. The self-impedance of the port closest to the IC is evaluated to minimize the self impedance of PDN, resulting in $p - 1$ ports being accessible for decaps placement.

Algorithm 1: Decap Optimization using Metaheuristic Optimization.

Input: $Z_{pdn} = Z_{p\times p\times f}$, $Z_{1\times f}$ (from manufacturer), Z_T, N, $MaxIt$, and the parameters
Output: Optimum capacitors and their port numbers, $Z_{obj} = (max(Z_{eq}(1, 1)))$.
Objective function: $Z_{obj} = max(Z_{eq}(1, 1))$, where $(Z_{eq})_f = (Z^{-1}_{pdn} + Z^{-1}_{Decap})^{-1}_f$, $\forall \in [0, f_{max}]$.
Initialize $n_D = 0$.
while $(Z_{obj} > Z_T)$ **do**
$n_D = n_D + 1$; $n = 0$;
Generate Initial population $X^{n=0}$
Compute initial best (minimum) $Z^{n=0}_{obj}$
while $(n < MaxIt)$**do**
$n = n + 1$;
  for $k = 1, 2, \ldots, N$ **do**
    for $d = 1, 2, \ldots, N_D$ **do**
      Update position $(X^n_{k,d})$ as per the algorithm
    end for
  Compute Z_{obj} for new position
  end for
Check current best (minimum) Z_{obj} and X_b at Z_{obj}
end while
Output: Z_{obj}, X_b = (Capacitor, Port)
end while

B. MATRIX-BASED EVOLUTIONARY COMPUTATION (MEC) APPROACH

Swarm-based metaheuristics for large-scale optimization problems may result in extremely high computing time. Due to the huge population, large-scale decision variables, or both. Designing parallel or distributed algorithms is a potential method to ease the computing strain while dealing with complicated optimization problems, solving large-scale optimization problems with the Matrix-Based Evolutionary Computation (MEC) [31] to reduce time complexity. MEC accelerates the processing speed by utilizing the matrix's parallel computing functionalities, the matrix operations [31]. This matrix operations such as addition (+), subtraction (–), multiplication (×), find maximization or minimization, and other linear operations have been widely studied in the parallel and distributed computing community. A vector is frequently used to represent an individual in metaheuristics, is

given as:

$$X_k = (x_{k1}, x_{k2}, \ldots, x_{kD}) \tag{17}$$

and a group of individuals (vectors) forms the population. In MEC, the whole population is represented as a matrix:

$$X = \begin{bmatrix} x_{11} & \ldots & x_{1D} \\ \vdots & \ddots & \vdots \\ x_{N1} & \ldots & x_{ND} \end{bmatrix} \tag{18}$$

where a row represents an individual and a column represents a dimension. So, the population X can be randomly initialized as:

$$X = ones \times lb + ones \times (ub - lb) \cdot rand_{N \times D} \tag{19}$$

where ub and lb are lower and upper bounds respectively, $ones$ is matrix of element 1, and $rand_{N \times D}$ is $N \times D$ matrix where each element is a random number $\in [0, 1]$. Here, $\cdot$ is scalar multiplication and $\times$ is matrix multiplication operation. Moreover, the global best solution can be found using matrix operation as:

$$[Z_{obj}, i] = \begin{cases} max(Obj(X)), & \text{for maximization problem} \\ min(Obj(X)), & \text{for minimization problem} \end{cases} \tag{20}$$

$$X_b = X_i \tag{21}$$

Population conduct position updates at individual and dimension levels throughout an iteration in the algorithm, eventually approaching the global optimum. For complete population, position update can be done at once using matrix operations, e.g. in M-PSO, position updates as:

$$V^{n+1} = w \cdot V^n + c_1 \cdot rand_{N \times D} \cdot (X^p - X^n) + c_2 \cdot rand_{N \times D} \cdot (ones_{N \times 1} \times X^g - X^n) \tag{22}$$

$$X^{n+1} = X^n + V^{n+1} \tag{23}$$

All the elements of the individuals in the population must lie between the lower and upper bounds. After updating the position, the boundary condition of whole population can be checked and updated using matrix operation at once, like for the lower bound condition:

$$Logic = X < (ones \times lb) \tag{24}$$

$$X = Logic \cdot lb + (1 - Logic) \cdot X \tag{25}$$

Here, $Logic$ matrix is $N \times D$ matrix which contains 0 and 1 only and helps to update X for the instant where the given condition (as in equation(25)) is true. Similarly, upper bound condition can be checked and updated.

In this way, MEC can alleviate the significant computational strain imposed by large population sizes and large-scale decision variables. MEC is a parallel and distributed algorithm on the population level, individual level, and also on dimension level. As a consequence, when MEC is combined with metaheuristics for large-scale optimization problems, the computing time is reduced.

Algorithm 2: Decap Optimization Using Matrix-Based Metaheuristics.

1: **Input:** $Z_{pdn} = Z_{p \times p \times f}$, $Z_{1 \times f}$ (from manufacturer), Z_T, N, $MaxIt$, and the parameters
2: **Output:** Optimum capacitors and their port numbers, $Z_{obj} = (max(Z_{eq}(1, 1)))$.
3: **Objective function:** $Z_{obj} = max(Z_{eq}(1, 1))$, where $(Z_{eq})_f = (Z_{pdn}^{-1} + Z_{Decap}^{-1})_f^{-1}, \forall \in [0, f_{max}]$.
4: Initialize $n_D = 0$.
5: **while** $(Z_{obj} > Z_T)$ **do**
6: $n_D = n_D + 1$; $n = 0$;
7: Generate Initial population $X^{n=0}$
8: Compute initial best (minimum) $Z_{obj}^{n=0}$
9: **while** $(n < MaxIt)$**do**
10: $n = n + 1$;
11: Update position X^n as per algorithm
12: Compute Z_{obj} for new position for whole population
13: Check current best (minimum) Z_{obj} and X_b at Z_{obj}
14: **end while**
15: Output: Z_{obj}, X_b = (Capacitor, Port)
16: **end while**

TABLE 2. Parameters Taken for Metaheuristic Algorithms

Algorithm	Parameters			
PSO	$c_1 = 1.5$	$c_2 = 1.5$	$w_i = 0.4$	$w_f = 0.9$
DE	$F = 0.5$		$pCR = 0.7$	
ABC	$N_o = N$		$L = 0.6 \times N$	
FTMA	$p = 0.7$		$r = 0.5$	

Algorithm-2 summarises pseudo code for the decoupling capacitor optimization methodology in a PDN using Matrix-Based Metaheuristics; that is, MEC is combined with metaheuristics. For matrix-based metaheuristics, the position update (also for each dimension) and objective function evaluation for each individual in the population occur in parallel, reducing the run time for each algorithm compared to the implementation of Algorithm-1. In Algorithm-2, there is no involvement of loops as the whole population is updated through the matrix operations at once unlike Algorithm-1 using loops (at an individual level, dimension level) for updating the population.

V. RESULTS AND COMPARATIVE ANALYSIS

For a fare comparison, for all the algorithms used for this work, the population size N is set to 20 and the maximum number of iterations $MaxIt$ is set to 50. Table 2 lists the parameters required to implement these algorithms. The algorithms were implemented in MATLAB R2019b and executed on a computer with 8 GB RAM and Intel i5 8^{th} Gen 2.4 GHz cores. For both the case studies, there were 10 independent

TABLE 3. Performance Comparison of the Metaheuristics for Case-Study 1

Criterion	PSO		DE		ABC		FTMA	
	A1	A2	A1	A2	A1	A2	A1	A2
n_{Avg}	5	5	10	8	6	6	4	4
n_{Dmin}	4	4	9	6	5	5	4	4
Z	58.9	57.7	55.3	59.6	56.1	58.7	56.0	55.5
T	362	272	917	437	1295	974	780	643

TABLE 4. Performance Comparison of the Metaheuristics for Case-Study 2

Criterion	PSO		DE		ABC		FTMA	
	A1	A2	A1	A2	A1	A2	A1	A2
n_{Avg}	8	8	9	9	9	9	5	5
n_{Dmin}	6	6	8	7	8	7	5	5
Z	59.1	54.5	57.1	55.5	56.8	56.7	55.8	53.1
T	572	368	686	421	1881	1231	942	744

TABLE 5. % Gain in Terms of CPU Time

Metaheuristic Algorithm	PSO	DE	ABC	FTMA
Case-study 1	24.86	52.34	24.78	17.56
Case-study 2	35.67	38.63	35.56	21.02

TABLE 6. Optimum Decoupling Capacitors and Port Numbers for Case-Study 1

Capacitor Index	Capacitor Model	Manufacturer	Port Number
7	C1005X5R0J475M050BC	TDK	6
2	GRT155R61E105KE01	Murata	10
2	GRT155R61E105KE01	Murata	15
3	CL05B104KO5NNN	Samsung	17

runs performed for all the algorithms to get an average estimate of the performance of each algorithm. The detailed results are not given due to the space constraints and the summary is given in this section.

Table 3 and Table 4 summarise the comparison of performance for Case-Study 1 and Case-Study-2, respectively, where n_{Avg} and n_{Dmin} indicates the average and minimum number of decaps, respectively, Z (in $m\Omega$) is the optimum impedance corresponding to n_{Dmin}, T (in sec) is the average CPU time, A1 and A2 corresponds to Algorithm-1 and Algorithm-2, respectively. Fig. 3 and Fig. 4 show critical results pictorially to get a quick idea about the performance comparison of these algorithms. It is evident that the MEC approach significantly reduces CPU time. This is summarised in Table 5. Fig. 5 and Fig. 6 shows the results for self impedance of PDN by each algorithm using minimum number of decaps. Table 6 and Table 7 shows the list of the optimum number of decaps with their index numbers (C) and corresponding port numbers (P) by FTMA.

Following are some of the observations about this specific optimization problem:

TABLE 7. Optimum Decoupling Capacitors and Port Numbers for Case-Study 2

Capacitor Index	Capacitor Model	Manufacturer	Port Number
1153	TPSB107M006R0250	AVX	6
1167	TPSB225K035R2000	AVX	7
2000	TRJB155K035RNJ	AVX	11
2003	TRJB156K010RNJ	AVX	15
2022	TRJB335K035RNJ	AVX	17

- For all the metaheuristic algorithms used in this paper, MEC based implementation is computationally very efficient than their conventional implementations. The maximum gain reported is upto 52% while the minimum one is 17%.
- The best-optimized impedance value and the minimum number of decaps is obtained by FTMA. Also, the average number of decaps obtained by FTMA is less as compared to other algorithms. ABC and DEA require more decaps, as illustrated in Fig. 3 and Fig. 4.
- The minimum average computation time (CPU time) was taken by PSO, and the ABC required maximum average computation time as compared to other algorithms.
- MEC-based algorithms have reduced the computation time (CPU time) significantly, as illustrated in Fig. 3 and Fig. 4.
- For the large dimensional problem in Case-study 2 (i.e. more decoupling capacitors are used for optimization), the percentage reduction in CPU time of the algorithms using MEC increases except in case of DE. This needs to be investigated further.

VI. CONCLUSION

This paper presents an approach for efficient selection and placement of decoupling capacitors in a power delivery network. The proposed approach use Matrix Evolutionary Computation (MEC) based techniques for optimization to improve the timing efficiency of metaheuristic algorithms. The MEC-based optimization approach proves to be more efficient than conventional metaheuristic algorithms. For this study, a practical power delivery network is considered, with the impedance being lowered by employing the fewest number of decoupling capacitors possible. In the context of MEC, a comparison of many prominent optimization algorithms (PSO, DE, ABC, and FTMA) is also provided. FTMA provides the best impedance with the fewest capacitors among the methods employed in this study, whereas PSO is the fastest approach for optimization.

REFERENCES

[1] R. Feynman, "There's plenty of room at the bottom," *Eng. Sci.*, vol. 23, no. 5, pp. 22–36, 1960.

[2] M. Swaminathan and E. Engin, *Power Integrity Modeling and Design for Semiconductors and Systems, Ser. Prentice Hall Modern Semiconductor Design Series*. Upper Saddle River, NJ, USA: Pearson Education, 2007.

[3] G. Boschetto, S. Carapezzi, and A. Todri-Sanial, "Graphene and carbon nanotubes for electronics nanopackaging," *IEEE Open J. Nanotechnol.*, vol. 2, pp. 120–128, Nov. 2021.

[4] V. K. Nishad, A. K. Nishad, B. K. Kaushik, and R. Sharma, "First-principle analysis of transition metal edge-passivated armchair graphene nanoribbons for nanoscale interconnects," *IEEE Trans. Nanotechnol.*, vol. 20, pp. 92–98, Jan. 2021.

[5] A. B. Amin and M. S. Ullah, "Mathematical framework of tetramorphic MWCNT configuration for VLSI interconnect," *IEEE Trans. Nanotechnol.*, vol. 19, pp. 749–759, Sep. 2020.

[6] G. Hills *et al.*, "Understanding energy efficiency benefits of carbon nanotube field-effect transistors for digital VLSI," *IEEE Trans. Nanotechnol.*, vol. 17, no. 6, pp. 1259–1269, Nov. 2018.

[7] J. Liang *et al.*, "Atomistic- to circuit-level modeling of doped SWCNT for on-chip interconnects," *IEEE Trans. Nanotechnol.*, vol. 17, no. 6, pp. 1084–1088, Nov. 2018.

[8] T. Song, "Opportunities and challenges in designing and utilizing vertical nanowire FET (V-NWFET) standard cells for beyond 5 nm," *IEEE Trans. Nanotechnol.*, vol. 18, pp. 240–251, Feb. 2019.

[9] Y. Yu and N. K. Jha, "Energy-efficient monolithic three-dimensional on-chip memory architectures," *IEEE Trans. Nanotechnol.*, vol. 17, no. 4, pp. 620–633, Jul. 2018.

[10] T. Pathade, Y. Agrawal, R. Parekh, and M. G. Kumar, "Structure fortification of mixed CNT bundle interconnects for nano integrated circuits using constraint-based particle swarm optimization," *IEEE Trans. Nanotechnol.*, vol. 20, pp. 194–204, Feb. 2021.

[11] M. Sanaeepur, "Crosstalk delay and stability analysis of MLGNR interconnects on rough surface dielectrics," *IEEE Trans. Nanotechnol.*, vol. 18, pp. 1181–1187, Oct. 2019.

[12] M. Sahoo, P. Ghosal, and H. Rahaman, "Modeling and analysis of crosstalk induced effects in multiwalled carbon nanotube bundle interconnects: An ABCD parameter-based approach," *IEEE Trans. Nanotechnol.*, vol. 14, no. 2, pp. 259–274, Mar. 2015.

[13] M. Rezaei Khezeli, M. H. Moaiyeri, and A. Jalali, "Analysis of crosstalk effects for multiwalled carbon nanotube bundle interconnects in ternary logic and comparison with Cu interconnects," *IEEE Trans. Nanotechnol.*, vol. 16, no. 1, pp. 107–117, Jan. 2017.

[14] E. Bogatin, *Signal and Power Integrity, Ser. Signal Integrity Library*. Upper Saddle River, NJ, USA: Prentice Hall, 2009.

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

[16] Z. Mu, "Power delivery system: Sufficiency, efficiency, and stability," in *Proc. 9th Int. Symp. Qual. Electron. Des.*, 2008, pp. 465–469.

[17] J. N. Tripathi, V. K. Sharma, and H. Shrimali, "A review on power supply induced jitter," *IEEE Trans. Compon. Packag. Manuf. Technol.*, vol. 9, pp. 511–524, Mar. 2019.

[18] L. Smith, R. Anderson, D. Forehand, T. Pelc, and T. Roy, "Power distribution system design methodology and capacitor selection for modern CMOS technology," *IEEE Trans. Adv. Packag.*, vol. 22, no. 3, pp. 284–291, Aug. 1999.

[19] T. Hubing, "Effective strategies for choosing and locating printed circuit board decoupling capacitors," in *Proc. Int. Symp. Electromagn. Compat.*, vol. 2, 2005, pp. 632–637.

[20] S. Piersanti, F. de Paulis, C. Olivieri, and A. Orlandi, "Decoupling capacitors placement for a multichip PDN by a nature-inspired algorithm," *IEEE Trans. Electromagn. Compat.*, vol. 60, no. 6, pp. 1678–1685, Dec. 2018.

[21] I. Erdin and R. Achar, "Multi-objective optimization of decoupling capacitors for placement and component value," *IEEE Trans. Compon. Packag. Manuf. Technol.*, vol. 9, no. 10, pp. 1976–1983, Oct. 2019.

[22] Z. Xu, J. Wang, and J. Fan, "Decoupling capacitor placement optimization with lagrange multiplier method," in *Proc. IEEE Int. Symp. Electromagn. Compat. Signal/Power Integrity*, 2020, pp. 22–25.

[23] 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.

[24] J. Wang, J. Lu, and Y. Li, "Placement of decoupling capacitor on packages based on effective decoupling radius," in *Proc. IEEE 18th Electron. Packag. Technol. Conf.*, 2016, pp. 195–198.

[25] H.-C. Kuo, C.-W. Kuo, C.-Y. Tsai, F.-C. Chu, and C.-C. Wang, "On-package decoupling capacitor performance improvement of flip-chip packages for high power application," in *Proc. 13th Int. Microsystems, Packag., Assem. Circuits Technol. Conf.*, 2018, pp. 245–247.

[26] J. N. Tripathi, R. K. Nagpal, N. K. Chhabra, R. Malik, and J. Mukherjee, "Maintaining power integrity by damping the cavity-mode anti-resonances' peaks on a power plane by particle swarm optimization," in *Proc. 13th Int. Symp. Qual. Electron. Des.*, 2012, pp. 525–528.

[27] 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.

[28] G. Han, "Simple and fast method of on-board decoupling capacitor selection and placement," in *Proc. IEEE Elect. Des. Adv. Packag. Syst. Symp.*, 2017, pp. 1–3.

[29] J. Chen and L. He, "Efficient in-package decoupling capacitor optimization for I/O power integrity," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 26, no. 4, pp. 734–738, Apr. 2007.

[30] P. Kadlec, M. Marek, M. Štumpf, and V. Šeděnka, "PCB decoupling optimization with variable number of capacitors," *IEEE Trans. Electromagn. Compat.*, vol. 61, no. 6, pp. 1841–1848, Dec. 2019.

[31] Z.-H. Zhan *et al.*, "Matrix-based evolutionary computation," *IEEE Trans. Emerg. Topics Comput. Intell.*, to be published, doi: 10.1109/TETCI.2020.3047410.

[32] J. N. Tripathi, P. Damle, and R. Malik, "Minimizing core supply noise in a power delivery network by optimization of decoupling capacitors using simulated annealing," in *Proc. IEEE 21st Workshop Signal Power Integrity*, 2017, pp. 1–3.

[33] 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.

[34] E. Kulali, E. Wasserman, and J. Zheng, "Chip power model - a new methodology for system power integrity analysis and design," *Proc. IEEE Elect. Perform. Electron. Packag.*, 2007, pp. 259–262.

[35] S. Kahng, "Ga-optimized decoupling capacitors damping the rectangular power-bus' cavity-mode resonances," *IEEE Microw. Wireless Compon. Lett.*, vol. 16, no. 6, pp. 375–377, Jun. 2006.

[36] X. Yang, "Introduction to mathematical optimization: From linear programming to metaheuristics," Cambridge Int. Sci. Publishing, 2008.

[37] J. Kennedy, R. Eberhart, and Y. Shi, *Swarm Intelligence*. San Francisco: Morgan Kaufmann Publishers, 2001.

[38] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. - Int. Conf. Neural Netw.*, vol. 4, 1995, pp. 1942–1948.

[39] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proc. IEEE Int. Conf. Evol. Comput. Proc. IEEE World Congr. Comput. Intell.*, 1998, pp. 69–73.

[40] 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.

[41] R. Storn and K. Price, "Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces," *J. Glob. Optim.*, vol. 11, pp. 341–359, 1997.

[42] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4–31, Feb. 2011.

[43] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm," *J. Glob. Optim.*, vol. 39, pp. 459–471, 2007.

[44] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, "A comprehensive survey: Artificial bee colony (ABC) algorithm and applications," *Artif. Intell. Rev.*, vol. 42, no. 1, pp. 21–57, 2014.

[45] Z. T. Allawi, I. Ibraheem, and A. J. Humaidi, "Fine-tuning meta-heuristic algorithm for global optimization," *Processes*, vol. 7, no. 10, 2019, Art. no. 657. [Online]. Available: https://www.mdpi.com/2227-9717/7/10/657

AKASH JAIN is currently working toward the Undergraduation degree in electrical engineering with the Indian Institute of Technology, Jodhpur, Jodhpur, India. His current research interests include analog and digital VLSI design, signal and power integrity, optimization and design techniques for circuits and systems.

HEMAN VAGHASIYA is currently working toward the Undergraduation degree in electrical engineering with the Indian Institute of Technology, Jodhpur, Jodhpur, India. His current research interests include digital VLSI design, signal and power integrity, metaheuristic optimization and soft computing techniques for designing circuits and systems, and machine learning techniques.

JAI NARAYAN TRIPATHI (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the Indian Institute of Technology Bombay, Mumbai, India, in 2014. He is currently an Assistant Professor with the Indian Institute of Technology Jodhpur, Jodhpur, India, and also an Adjunct Research Professor with Carleton University, Ottawa, ON, Canada. In past, he was with STMicroelectronics, India, for almost seven years, where he was involved in the design issues of various high-speed Systems-on-Chip (SoCs). In 2016 and 2017, he was a Visiting Scientist with the Politecnico di Torino, Turin, Italy, where he was also a Visiting Postdoctoral Fellow in 2016. He has authored or coauthored more than 75 research papers in refereed journals, as book chapters and in the proceedings of top international conferences. His current research interests include signal integrity, power integrity, electromagnetic interference or electromagnetic compatibility, metaheuristic optimization, and RF circuits. He is awarded a grant from the Department of Science and Technology, Govt. of India, to work on a power integrity problem.

Dr. Tripathi was a TPC Member for more than 20 international conferences, including the premier conferences, such as IEEE EPEPS, IEEE VLSI Design, and IEEE EDAPS. He was the recipient of the Young Investigator Training Program Research Award by ACRI, Italy, in 2016 and 2017, consecutively. He was an Invited Speaker in IEEE EDAPS 2015, Seoul, South Korea, where he was also a Session Co-Chair of the session High-Speed Channels and Interconnects. He was a TPC Co-Chair of IEEE EDAPS 2018. He was a reviewer for many international journals, such as IEEE TCAS-I: Regular Papers, IEEE TCAS-II: Express Briefs, IEEE TVLSI, IEEE TEMC, IEEE TPES, PIER, and *Microelectronics Journal*. He has delivered invited talks at various universities and international forums. He is a Member of a technical committee TC-12 EDMS of IEEE EPS Society. Dr. Tripathi is currently an Associate Editor for the IEEE OPEN JOURNAL OF CIRCUITS AND SYSTEMS and IEEE TRANSACTIONS ON COMPONENTS, PACKAGING AND MANUFACTURING TECHNOLOGY.