electronics

MDPI

Article

Effective PCB Decoupling Optimization by Combining an Iterative Genetic Algorithm and Machine Learning

Riccardo Cecchetti [1], Francesco de Paulis [1], Carlo Olivieri [1], Antonio Orlandi [1,*] and Markus Buecker [2]

[1] UAq EMC Laboratory, Dept. of Industrial and Information Engineering and Economics, University of L'Aquila, 64100 L'Aquila, Italy; riccardo_cecchetti@fastwebnet.it (R.C.); francesco.depaulis@univaq.it (F.d.P.); carlo.olivieri@univaq.it (C.O.)

[2] Zuken GmbH, 33104 Paderborn, Germany; markus.buecker@zuken.com

* Correspondence: antonio.orlandi@univaq.it; Tel.: +39-0862-434432

Received: 20 June 2020; Accepted: 31 July 2020; Published: 2 August 2020

Abstract: An iterative optimization for decoupling capacitor placement on a power delivery network (PDN) is presented based on Genetic Algorithm (GA) and Artificial Neural Network (ANN). The ANN is first trained by an appropriate set of results obtained by a commercial simulator. Once the ANN is ready, it is used within an iterative GA process to place a minimum number of decoupling capacitors for minimizing the differences between the input impedance at one or more location, and the required target impedance. The combined GA–ANN process is shown to effectively provide results consistent with those obtained by a longer optimization based on commercial simulators. With the new approach the accuracy of the results remains at the same level, but the computational time is reduced by at least 30 times. Two test cases have been considered for validating the proposed approach, with the second one also being compared by experimental measurements.

Keywords: machine learning; artificial neural network; decoupling capacitors; power delivery network; genetic algorithm; twin removal; binary coding; power integrity; signal integrity

1. Introduction

The Artificial Intelligence (AI) paradigm is pervading the entire technical word, from humanities, social sciences, to medicine (as the recent COVID-19 scenario has shown) to, of course, engineering.

In this field, one of the last (most recent) areas to be approached by this paradigm is the design of electronic hardware. The reason is the complexity of the problem, the huge variety of the domain of application, the interlaced design rules and/or design strategies [1–4]. In particular, AI comes to help the design and optimization of printed circuit boards (PCB) in terms of layout, component placement, interconnect routing, channel and equalization optimization [5,6]. A key aspect in the PCB design of modern electronic systems dealing with high speed digital signals and lower DC voltages is the huge demand of current required by the ICs. Such a current generates a high frequency voltage ripple overlapped to the DC level that, in turn, may degrade the digital signaling [7–9]. Proper decoupling strategies should be adopted at the chip, package and PCB level to ensure that the induced voltage noise is maintained within a specified range [10–15]. To this aim, optimization algorithms have been shown to effectively solve this problem [16–19] by achieving the input impedance seen at the power input terminals of the chip (or package) to be lower than the target impedance set by the voltage noise constrains in time domain.

There are several types of equivalent circuit models depending on their complexity and corresponding accuracy that can be used for quickly evaluating the input impedance of the power

delivery network (PDN) [20–24]. However, nowadays most of designers have, together with the commercial layout tools for PCB design, side software able to run power integrity simulations, and thus to obtain the input impedance based on the design PDN. Although this commercial software is often available, their setup and calculation time can be often a drawback. The Genetic Algorithm (GA) optimization described in [19] has been shown to be effective for predicting the PDN input impedance for an effective pre-layout placement optimization of decoupling capacitors. The GA in [19] relies, for the PDN impedance calculation, on the commercial solver [25] with some limitation such as the possibility to run at each generation the input impedance calculation once the layout is already set, thus using defined types (packages) of decaps and its inherent long computation time.

The target of this paper is to use a Machine Learning (ML) approach based on Artificial Neural Networks (ANN) to substitute the use of the commercial solver in [19]. The ANN, once appropriately trained, can quickly predict with a good accuracy the input impedance for a given design.

The presently proposed ANN approach overcomes several limitations of the commercial solver-based optimization proposed in [19] where the GA optimization was effective as long as all decaps had the same package. However, this greatly limits the possibility to span a usually larger decap database consisting of different types of decap packages. Once the parasitic inductance and resistance of each package associated to their corresponding placement (pads size, vias diameter, etc.) are evaluated, they can be embedded in the decap ESL and ESR and be taken into account by the ANN simulation. Therefore, once the ANN training is accomplished based on a simplified layout (only one package type considered), the ANN is much more effective and quick to provide the best solution at a pre-layout PDN design stage.

The obtained impedance profiles from the ANN while running the iterative optimization algorithm are compared to the corresponding impedances evaluated by a commercial tool and to experimental data. Good agreement is found among these curve sets, validating the effectiveness and accuracy of the proposed ML-based decap placement.

The present work is structured as follows: Section 2 introduces the architecture of the ANN implemented into the optimization process, and its features. In Section 3, the basic structure of the GA is recalled, and the changes for its used in combination with the ANN are highlighted. Section 4 is devoted to illustrating the physical structure and the electrical properties of the PDN. In the same section, the results of the optimization processes are also presented and validated through the corresponding experimental data and simulation response by a commercial simulator. Section 5 draws the final conclusions.

2. The Machine Learning Approach

The artificial feedforward neural networks (FF-ANN, in the following ANN) are algorithms suitable for the approximation of continuous functions on compact domains. Such a function can be approximated, within a desired accuracy, by an ANN with a given number of inner layers and neurons. In the present work, the regression capability of an ANN is exploited inside a genetic optimization algorithm (GA) as computational engine for the cost function. In [26], an approach for the optimization of the input impedance ($Z_{in}(f)$) of a power delivery network (PDN) has been described, simultaneously evaluated at multiple ports (i.e., 4) on the PDN itself.

By means of an iterative GA [19], an optimal configuration (position and value) of decoupling capacitors (or decaps) has been found, that simultaneously minimize, in a given frequency range, the $Z_{in}(f)$ at the ports Pi i = 1,4 associated to four fixed locations on the PDN. As a computational engine for the cost function of the GA, the commercial power integrity (PI) simulator Design Force (DF) has been used [25]. The GA approach starts from a randomly selected population of chromosomes. Each chromosome contains genes; each gene represents the relevant information of a decap—its position and value. To each chromosome, a cost (evaluated by the user defined cost function [19]) is assigned, that is a function of the frequency spectrum of the magnitude of the input impedance (also named the impedance profile). A procedure selects the chromosomes that have the best fitness

(in our case it corresponds to the lowest cost) and uses these chromosomes to create the next generation of the population by the known selection, breeding and mutation steps [27]. After a suitable number of generations, the GA converges to the global minimum value of the cost function and at its associated chromosome.

At the base of this contribution, there is the idea to substitute a trained ANN in the GA flow to the commercial PI simulator, that when given a chromosome (position and values of decaps) as input, is able to compute $Z_{in}(f)$. The approach described in [19] implies long computational times due to the repetitive use of the PI simulator for the computation of $Z_{in}(f)$ at each iteration. A trained ANN gives the same results in an infinitely shorter time. Of course, the computational time gain during the evaluation of $Z_{in}(f)$ by the ANN are compensated by the more computationally intense training phase. This phase is performed once off-line, and it is not included in the iterative process of the GA. On the same computing platform, the time saved by using the ANN is a few minutes, versus several hours when using the classic PI simulator.

2.1. The Artificial Neural Network Structure

The standard fitting problem of a function can be simplified and stated as follows: given a set of pairs <In(i),Out(i)> with i = 1,N, such that Out(i) = g(In(i)), to find the function h(.) that better approximates g(.). In the frame of the ANN, the set <In(i),Out(i)> is the training set (TS) from which the ANN "learns" and builds h(.). Then, h(.) is used to compute new output Out(j) = h(In(j)) due to input values In(j) not belonging to the training set. This property of the trained ANN is called "generalization".

The problem at hand calls for the fitting of a family of functions (the input impedances of the PDN) depending on the frequency f. For a specific input (a chromosome) the ANN should compute the entire $Z_{in}(f)$ in a specified frequency range.

The TS has the number, location on the PDN and value of the decoupling capacitors as input, and the frequency spectrum of the magnitude of $Z_{in}(f)$ at each of the four ports for that specific input as target (or label). The TS is computed using the PI simulator. It is worth pointing out that during the iterative process of the GA, the $Z_{in}(f)$ at the ports will be computed (for each of the chromosomes generated by the GA) by means of the trained ANN.

It is not trivial to decide the architecture of an ANN a priori based on its application. The key point is to set the number of inner layers and the number of neurons per layer [28]. If the complexity of the ANN is small (the number of layers and neurons is reduced), the fitting performance is also reduced. If the complexity is too high with respect to the order of the function to be fitted, then the ANN will overfit the data [29]. In our case, we have used the cross validation technique using independent training and test sets as reported in [28]. Starting from an architecture with three inner layers, chosen among the benchmarks in [29], several training procedures have been run, varying the number of neurons of each layer in turn and recording the performance of mean squared error (MSE) of the trained ANN in term. In the end, the architecture associated to the minimum MSE will have been considered and retained. In order to save time, this activity only aimed to set the most suitable ANN architecture, and the PI simulator has not been used but rather a hybrid equivalent circuit model of the PDN, in which either the lumped electrical behavior of the decaps and the distributed electrical behavior of the planes [20] are considered. Figure 1 shows the comparison between the frequency spectrum of the magnitude of the input impedance computed by the hybrid equivalent circuit (hybrid eq. ckt) and the same impedance computed by the PI simulator (PI sim.).

After this cross validation phase, the final structure of the ANN (shown in Figure 2) consists of 3 inner hidden layers (plus the input and output one) with 2 neurons each in the first and third, and 30 neurons in the second. The ANN has 20 inputs and 361 outputs (the values of $Z_{in}(f)$ at the 361 frequency points of the spectrum).

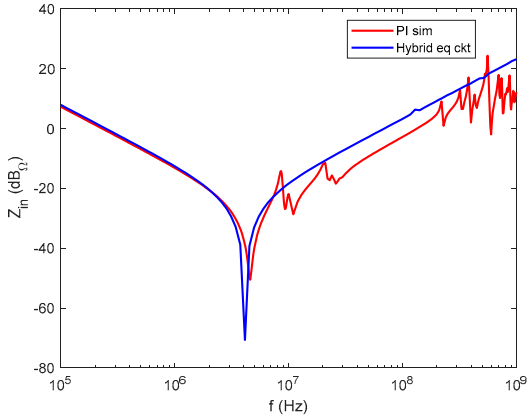


Figure 1. Frequency spectrum of the magnitude of $Z_{in}(f)$ computed by the hybrid equivalent circuit and by the PI simulator (PI sim.).

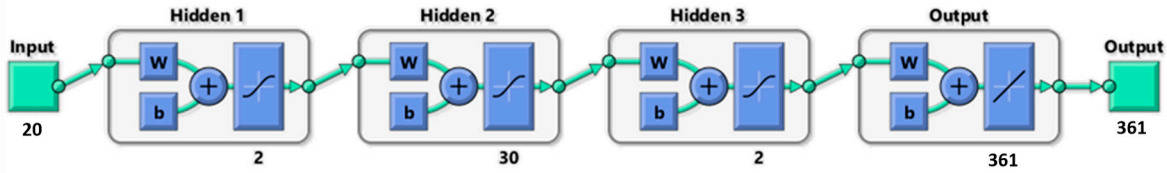


Figure 2. Artificial neural network (ANN) architecture.

The target of the GA is the minimization (below a user defined mask) of the $Z_{in}(f)$ of the board evaluated in four distinct locations. The evaluation of the four input impedances is obtained by four ANNs with the same architecture (as discussed above and in Figure 2), but trained with different training sets.

2.2. The Training Set

The ANN training starts with the definition of the TS. In order to optimize the computing resources, simple coding is used to represent the relevant information of the decaps. Three types of decaps are used and coded with numerals 1, 2 and 3, as reported in Table 1.

Table 1. Decoupling capacitors and their code.

Type	Code	C, ESL, ESR
1	1	100 nF, 222 pH, 8.9 mΩ
2	2	47 nF, 154 pH, 21.4 mΩ
3	3	22 nF, 142 pH, 25.2 mΩ

These decoupling capacitors are commercial components [30] specifically designed for decoupling due to their very low equivalent stray inductance (ESL). The placement of the decaps on the board is constrained to a grid of fixed positions, as shown in Figure 3. The number of allowable positions is 52. The positions are coded by a position designator (PD) ranging from 1 to 52.

In order to speed up the generation of the TS, a configuration matrix is built. It will be used as input to the PI simulator to compute the frequency spectrum of the $Z_{in}(f)$ at the four ports in Figure 3 in a specified frequency range for any given decap configuration (number, position and value). The structure of the input matrix (the dimensions of which are 20×2466) is reported in Figure 4.

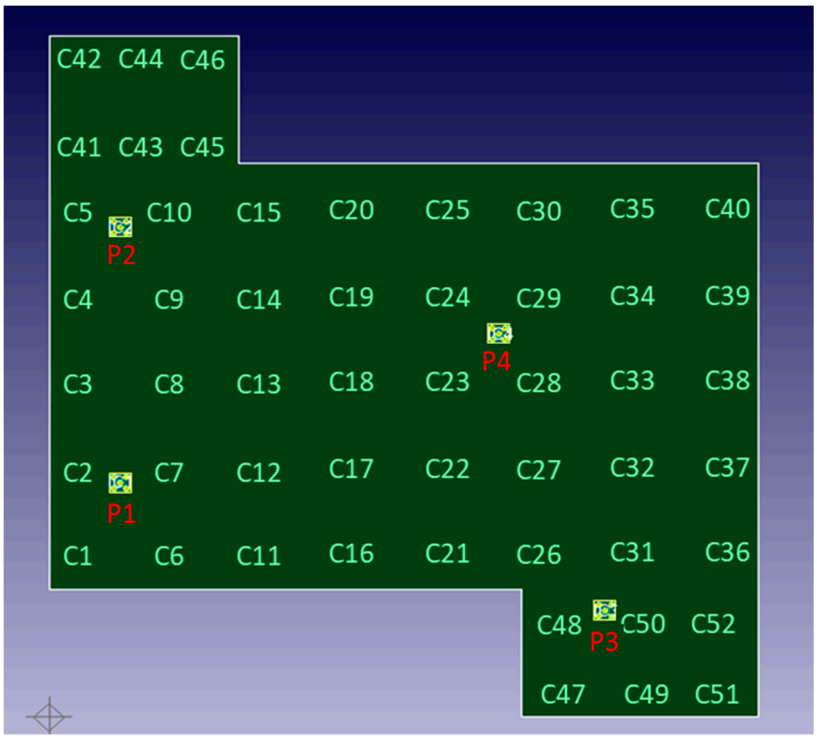


Figure 3. Schematic of the power delivery network (PDN) board: position of the 4 ports (P1 to P4) and grid of the allowed positions of the decaps.

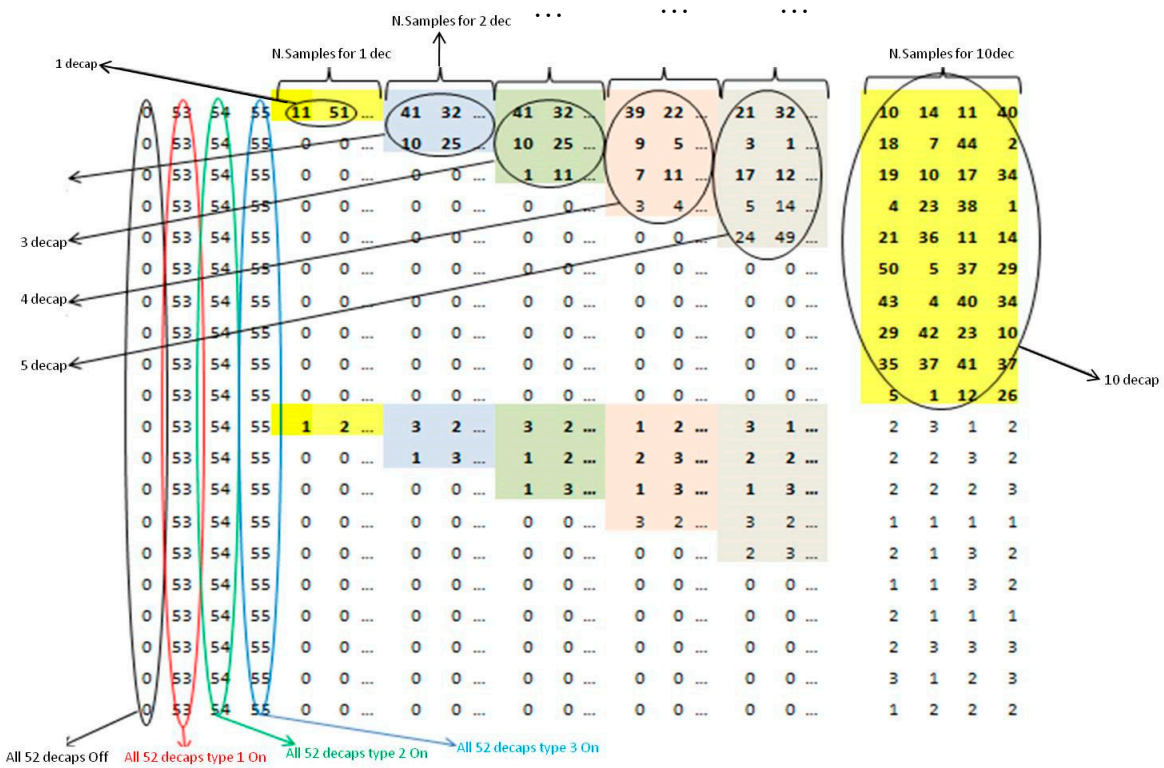


Figure 4. Structure of the input matrix for the generation of the training set (TS).

Each column represents one of the training configurations.

The first 10 rows (1 to 10) are dedicated to identifying the position of the decaps on the virtual grid of the board by means of the PD. In the last 10 rows (11 to 20), the type of decap is reported by

means of the coded 1 to 3 decap typology: in each column the decap with PD in line 1 has the type in line 11, the decap with PD in line 2 has the typology in line 12 and so on.

The TS should be not only representative of the most significant configurations of the actual working space, but should also be matched with the iterative nature of the GA. As reported in [19] the developed GA uses an iterative scheme: it considers one decap at a time. There are configurations with 1 to 10 decaps. Thus, the ANN should be trained with all these configurations. In the input matrix of Figure 4, there are columns considering configurations with only 1 decap on the board, and columns considering only 2 decaps on the board up to the case of 10 decaps. These sets of columns are shaded with different colors in Figure 4. The values of the PD and of the type of capacitor are randomly generated.

Given the number of possible decap locations on the virtual grid of the PDN board, the configurations (location and value) of decaps dramatically increases with the increasing number of decaps effectively considered on the board. Because of this, the number of configurations considered in the input matrix of the TS it is dependent upon the number of decaps, ranging from 60 for the case of only one decap on board, to 600 for the case of ten decaps.

As the number of decaps increases, the corresponding number of their possible combinations increases, and, consequently, the number of configurations used to train the ANN should also increase.

A limit of the ANN is its reduced capability to generalize the outputs for inputs that are outside the boundaries of the input configurations used in the training set. To overcome this limit, the TS also includes some "limit" input configurations as described in the following list:

- Bare board (no decoupling capacitors at all), column 1 of matrix in Figure 4.
- All decaps of type 1, column 2 of matrix in Figure 4.
- All decaps of type 2, column 3 of matrix in Figure 4.
- All decaps of type 3, column 4 of matrix in Figure 4.

Furthermore, in order to improve the prediction performances of the ANN and extend its training domain, 22 configurations previously obtained by the application of the GA in [19] have been added to the input matrix of the TS as the last 22 columns.

As already mentioned, the output matrix of the TS has been generated by using the PI simulator. For each input configuration contained in the input matrix, the PI simulator computes the frequency spectrum of the magnitude of the input impedance at the specified Port. The spectrum of $Z_{in}(f)$ consists of 361 values corresponding to the same number of frequency points logarithmically separated in the frequency range from 100 MHz to 1 GHz. As an example, Figure 5 shows one of the full training sets (2466 spectra of $Z_{in}(f)$ at one port).

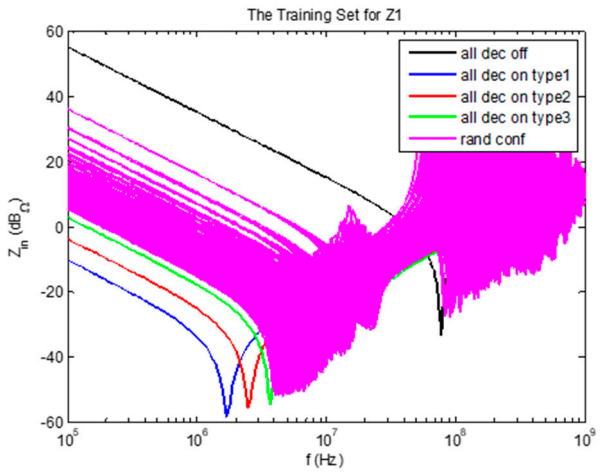


Figure 5. TS for one of the ports (Port 1).

After some tests, the algorithm chosen for the training is the Bayesian Regularization algorithm [31] for its properties of generalization. This property is very important because during the GA procedure the chromosomes that are generated are not included in the input training set (although inside its boundaries) and the ANN should compute the associated output with maximum accuracy. In each of the four TSs used to train the 4 ANNs for the 4 ports, 70% of the output has been used for the training, 15% for the validation, and the remaining 15%—not included in the training set of data—for the test check [32]. Figure 6 shows how, during the training, the proposed ANN converges very rapidly (in less than 100 epochs) and steadily to low values of MSE for both the training (blue line in Figure 6) and the test (red line in Figure 6) sets. The MSE of the test set reaches a minimum at an epoch number similar to that at which the MSE of the training (and associated validation) set reaches the minimum; thus also confirming a correct subdivision (70%, 15%, 15%) of the TS. From Figure 6, it is evident as the MSE of the test set drops from about 5000 to 1.7 in less than 100 epochs, and then remains still (although not shown) up to 1000 epochs.

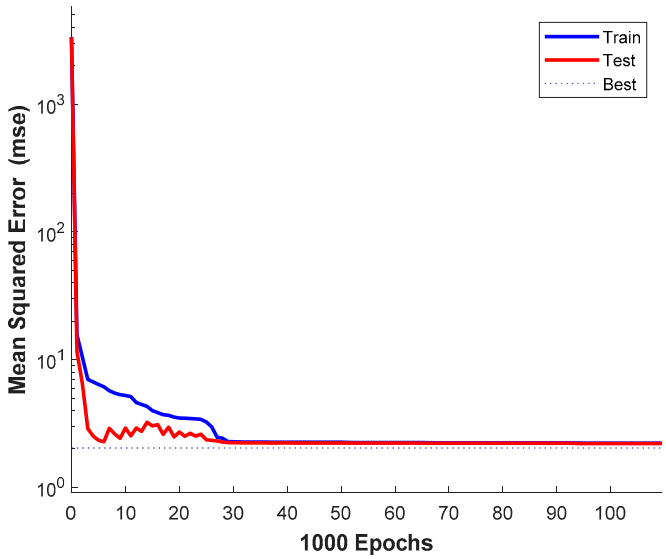


Figure 6. Training performance of the ANN.

2.3. ANN Validation

The developed ANN is validated based on several test cases belonging to the TS. One of these cases is reported in Figure 7 and compared to the target impedance profile obtained using the accurate PI simulator. Good accuracy is found between the two impedance curves. At low frequencies, below the first resonance (at around 9 MHz), the ANN output matches the capacitive behavior of the board very well. Above this resonance the inductive trend of the impedance is also well predicted, showing an increase in the impedance magnitude of about 20 dB/dec. The sharp anti-resonance between 10 and 20 MHz is missed, probably due to the numerically low-pass filtering effects due to the Bayesian Regularization algorithm [31] that introduces a higher damping (a lower quality factor or Q-factor) in the results. This is also noted above 100 MHz, when the electromagnetic distributed behavior of the considered board begins: beyond this frequency, the ANN output matches the frequency values of the resonances but shows lower values of amplitude. These accuracy considerations in conjunction with the very small computing time used to get the results make the proposed ANN a good candidate to replace the commercial PI simulator in the GA-based decap optimization process. In Figure 7, the MSE between the target curve and the ANN output is 1.212.

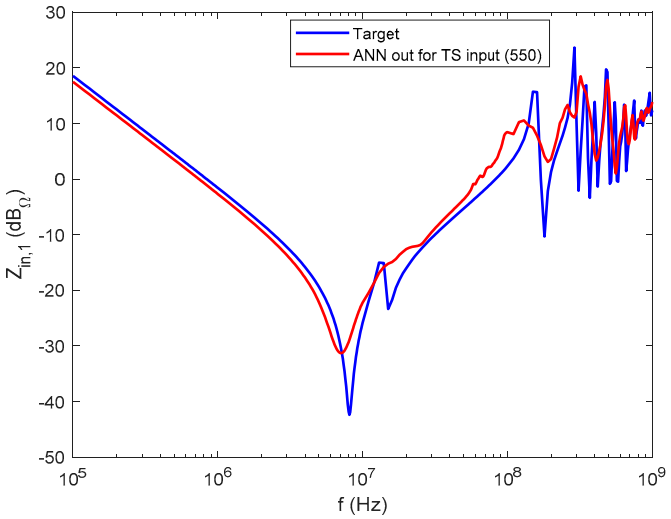


Figure 7. Comparison between the ANN output and the PI simulation results of the impedance at Port 1 for one decap configuration belonging to the TS.

3. The Genetic Algorithm

The architecture of the GA is similar to the one proposed in [19]. The GA iteratively places one decap at each optimization run, such that when the exit condition from the algorithm is met, a minimum number of decap is obtained. The GA is based on a binary encoding of the chromosome composed by two variables, as shown in Figure 8. The first variable corresponds to the position designator (PD), that represents the number of the locations on the PCB surface where the decaps can be placed; this number, with the corresponding *x*-*y* coordinates, is set by the user, according to the layout constraints (in our case can range from 1 to 52). The second consists of the specific decap typology to be used, with such decaps belonging to the group of in-the-shelf available decaps (in our case range from 1 to 3 according to Table 1). As shown in Figure 8, each element of the chromosome is converted into a binary number with a $N_b = 7$ bits encoding, thus having $N_c = 2^{Nb} = 128$ configurations available. The total possible positions are $N_{pos} = 52$; thus, the total N_{pos} locations are divided into N_c groups of size D_q according to Equation (1). The binary number obtained by the GA is first converted into its decimal counterpart Pos_{dec} using the typical ADC algorithm [33], and then the corresponding position is computed using Equation (2).

$$D_q = \frac{N_{pos} - 1}{2^{Nb} - 1} \tag{1}$$

$$Pos = floor\left(1 + D_q \cdot Pos_{dec}\right) \tag{2}$$

where the function *floor* rounds to the smaller integer number. The position coding leads to a $D_q = 0.4016$ using Equation (1); thus, the binary number '1101110' in the example in Figure 8 is first converted into $Pos_{dec} = 110$, and then to $Pos = 45$ using (R1.2.b) that is used to identify the decap position into the PCB. The same process can be applied to the coding of decap topology; by using $N_{dec} = 3$ in Equation (1) leading to $D_q = 0.0157$. The binary number '1111111' in the example in Figure 8 is first converted into $Cap_{dec} = 127$, and then into $Cap = 3$ using Equation (2). The conversion from the position (and decap type) into a binary number appropriate for the GA process is performed by inverting Equation (2) as done in Equation (3), where the function *ceil* rounds to the next integer. The specific case reported in Figure 8 for the conversion of the position $Pos = 45$ into its binary counterpart applies Equation (3) to get $Pos_{dec} = 110$. The corresponding binary number coded by 7 bits is '1101110'. The same procedure can be readily applied to the decap type coding.

$$Pos_{dec} = ceil\left(\frac{Pos - 1}{D_q}\right) \tag{3}$$

Figure 8. Structure of the chromosome and 7-bit binary encoding.

Each allowable decap is characterized by its package, its capacitance value and the corresponding parasitic inductive and resistive parameters ESL and ESR, respectively. The ESL and ESR should also embed the contribution of the mounting inductance and resistance for each different package type, if different from the single package type considered during the ANN training.

The flowchart of the iterative optimization based on the ANN calculation is reported in Figure 9. The main features of the algorithm are the following:

- The population size N_p;
- The number of generations N_{gen};
- The binary coding that is able to better randomize the generation of new chromosomes, since it is based on a limited number of integer values associated to both locations and decap type;
- The twin removal aimed at avoiding any redundancy in the creation of new chromosomes within a given population;
- The evaluation of the cost function f_{cost}, that, in general, can be associated to multiple input impedances as the sum of the cost functions f_{cost_i} of the N_{ports} for which the PDN impedance should be optimized, with $i = 1\ N_{ports}$.

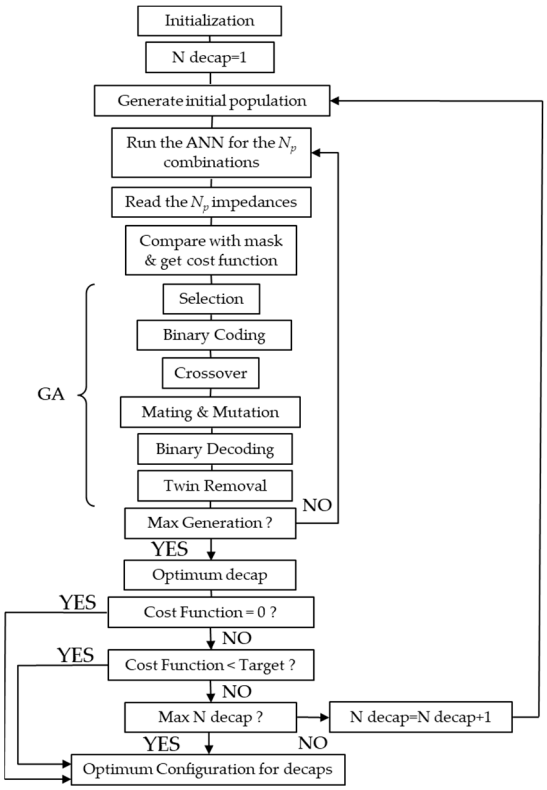


Figure 9. Flow chart of the iterative genetic algorithm (GA) optimization based on the ANN impedance calculation.

The overall cost function and the one associated to each port are computed according to Equations (4) and (5).

$$f_{cost} = \sum_{i=1}^{N_{ports}} f_{cost\_i} \tag{4}$$

$$f_{cost\_i} = \frac{\sum_{j=1}^{N_{freq}} \Delta Z_{in\_i}(f_j)}{N_1} \tag{5}$$

where the frequency range is made by N_{freq} frequency points. Each cost function f_{cost_i} increases by Z_{in_i} whenever the computed $Z_{in\_i}(f_j)$ is larger than the target $Z_{mask}(f_j)$, as described in Equation (6).

$$\Delta Z_{in\_i}(f_j) = \begin{cases} \left|Z_{in\_i}(f_j) - Z_{mask}(f_j)\right| \text{ when } \left|Z_{in\_i}(f_j)\right| \geq \left|Z_{mask}(f_j)\right| \\ 0 \text{ when } \left|Z_{in\_i}(f_j)\right| < \left|Z_{mask}(f_j)\right| \end{cases} \tag{6}$$

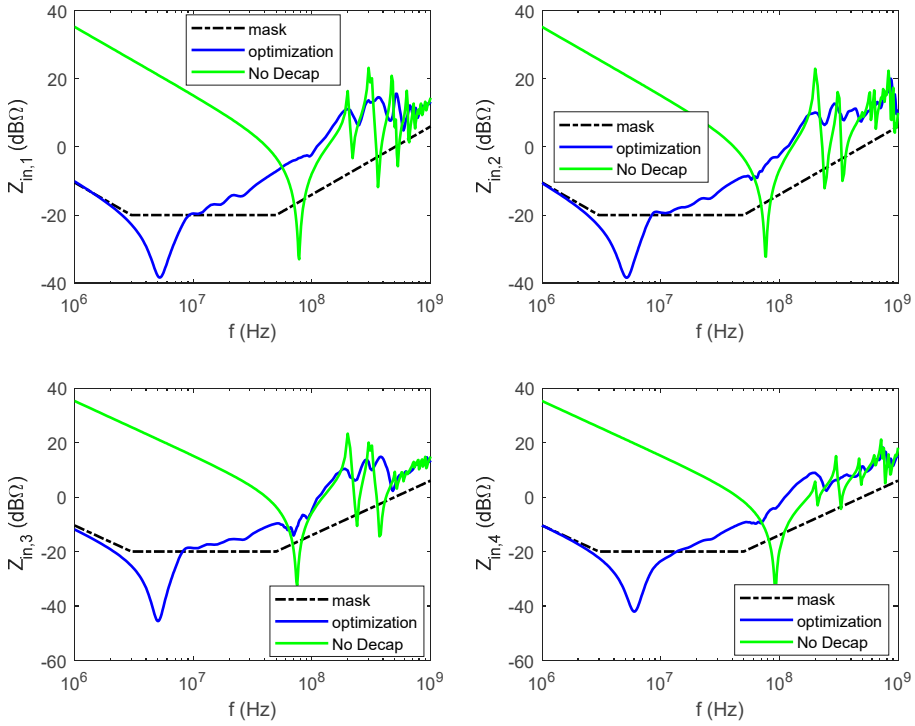
The optimization is considered accomplished based on three sub-conditions that can be set by the user according to the design specs: the first occurs when the cost function reaches zero, thus when the optimized impedance is below the target impedance on the whole frequency range considered. The second forces the optimization to stop when a user-defined target value of the cost function is reached. The third is based on a maximum number of decaps, corresponding to the number of iterations, to be placed on the board.

The GA implements its main steps (selection, crossover and mutation) on binary-coded chromosomes; also, the cross-over is randomly selected when generating each new chromosome. Such features greatly improve the randomness of the explored variable domain.

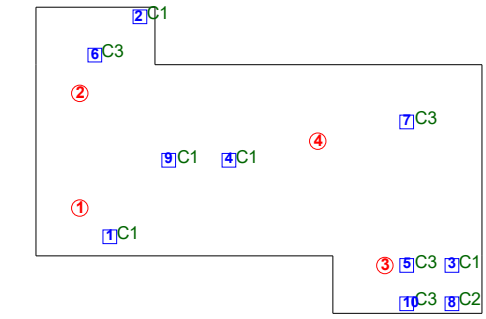
4. Numerical and Experimental Results

4.1. GA Optimization Based on ANN

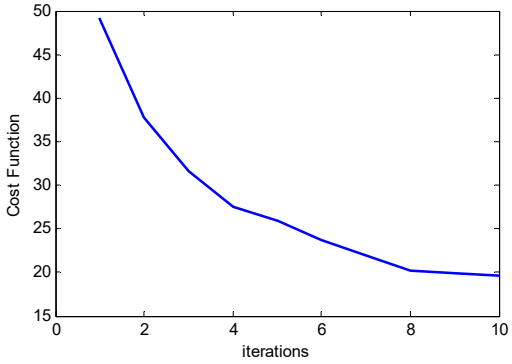
The developed ANN is employed in this section, together with the GA optimization process described in Section 3, to find out the best decap configuration that makes the input impedances at Ports 1–4 lower than the target profile (the mask indicated in Figure 10). The ANN described in Section 2 was trained to optimize the placement of 10 decaps. The optimization results of the impedances at the four ports are shown in Figure 10a; such curves correspond to the decap placement in Figure 10b, where the red circles and the numbers therein identify the ports. The blue squares identify the decap locations, whereas the numbers from 1 to 10 correspond to the order with which each decap has been placed during the 10 optimization cycles. The notation C_i, for i = 1, 3 stands for the typology of capacitors that were found as the optimum type at the given iteration. The cost function that is evaluated at each of the 10 iterations is shown in Figure 10c; it is characterized by the expected decreasing trend. Once the low frequency capacitive portion of the impedances reaches and goes below the target impedance given by the mask, the cost function decreases more slowly since further optimization can only slightly bring down the inductive portion of the impedances. Although not reported as curves, for sake of clarity, in Figure 10a a monotonic trend of the ANN output (the frequency spectrum of the port input impedance) has been noted toward the lower values of its magnitude as the iteration count increases (see Figure 10c). Because of this, the frequency spectrum of the magnitude of the port impedance associated to the last iteration has been considered as the final result. It should be pointed out that the random nature of the GA calls for multiple executions of the optimization process in order to perform a statistical analysis of the outputs in order to identify the best result. Because the target of this work was enough to obtain the first and quick fulfillment of the input impedance mask limit in the range of effectiveness of the decaps (around 10^7 Hz), only one execution of the proposed algorithm (of course with multiple internal iterations) was run.



(**a**)



(**b**)



(**c**)

Figure 10. Optimization results after running the combined ANN-GA process for the placement of 10 decaps. (**a**) input impedances at the four ports, (**b**) decap placement with the identification of the capacitor type listed in Table 1, (**c**) Decreasing trend of the cost function.

4.2. GA–ANN Optimization and Experimental Validation

A further GA–ANN based optimization is run in this section aimed at supporting and confirming the usefulness of the proposed combined procedure and at validating experimentally the obtained decap configuration.

The PCB briefly introduced in Figure 3 is manufactured and tested. The board dimensions and stack-up are reported in Figure 11. A second optimization run is performed based on the GA algorithm described in Section 3 that places one decap iteratively by searching for the best value and location among all the available positions identified in Figure 11a. At each iteration, the GA calls $N_p \cdot N_{gen}$ times the ANN developed in Section 2, with $N_p = 10$ and $N_{gen} = 10$. The optimization process is much quicker when using the ANN rather than the direct DF simulator, taking 0.7 s and 30 s for each impedance calculation, respectively.

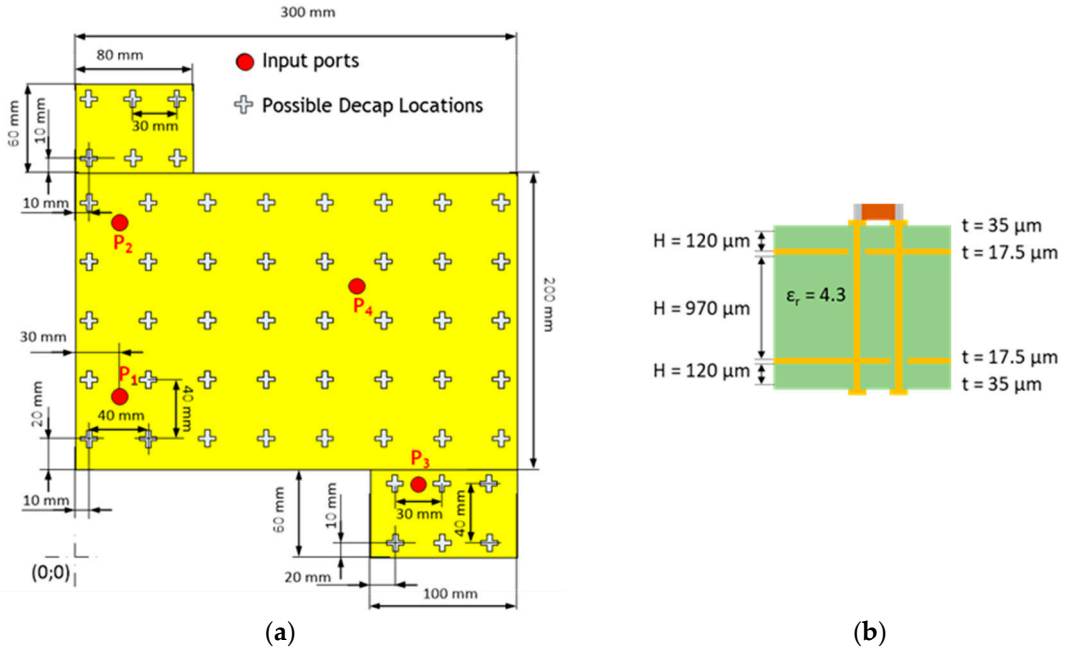
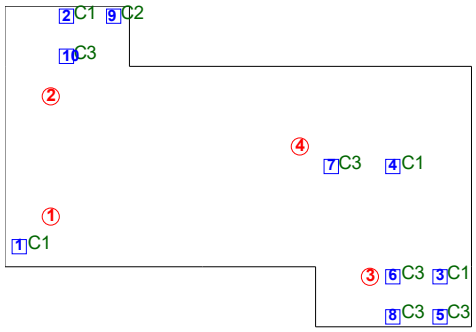


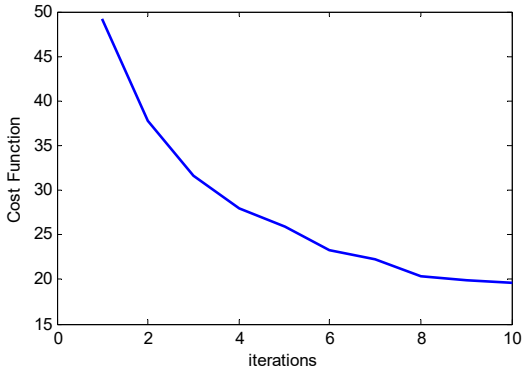
Figure 11. (**a**) The test PDN boards and (**b**) its stack-up.

This second GA–ANN based optimization outputs the decap configuration in Figure 12a. The first four decaps are of type 1 (C_1); this can be explained by the need to lower first the low frequency capacitive portion of the impedance; such decap placement will provide a steeper reduction in the cost function shown in Figure 12b. Subsequently, the C_3 decaps are inserted from the 5th to the 8th iteration. Finally, one C_2 and one more C_3 decaps are provided. The placement results in Figure 12a are similar to the layout results in Figure 10b, thus confirming the validity and consistency of the developed GA–ANN process. However, since the input impedance calculations by the ANN are based on configurations that are randomly selected by the GA within the same population and at each generation, the best solution at each iteration (for the placement of each new decap) may not be unique.

The decaps are mainly placed by the GA close to the ports, thus confirming that the placement inductance plays a role in the optimization process. Such inductance is accurately taken into account by the DF simulator; however, it is well predicted by the ANN, since the decap placement computed by the ANN proposed in this paper and shown in Figure 12a is consistent with the results obtained directly by the DF simulator in [19,34].



(**a**)



(**b**)

Figure 12. Application of the ANN within the GA optimization process. (**a**) Optimized decap configuration and placement. (**b**) Decreasing trend of the cost function.

The decap placement in Figure 12a is used for assembling a test board according to the geometry in Figure 11. The PCB with the identification of the decaps is shown in Figure 13. The measurement results are reported in Figure 14 for two of the four measured input impedances at Port 1 and 3. The measured impedances are compared to the impedances predicted by the ANN output at the final tenth iteration and to the accurate calculation provided by the DF simulator. The mask and the impedance of the bare board are also included. The measured impedance agrees well with one obtained from the DF simulation. In accordance with that already mentioned in Section 2.3, in this experimental case, the impedance predicted by the ANN shows a general agreement with the simulated and measured data. In the capacitive region (frequencies below the first resonance) the agreement is good. Moreover, in this case, the sharp anti-resonance following the first resonance at 9 MHz is not captured by the ANN. From 10 MHz to the beginning of the electromagnetic distributed behavior (around 200 MHz), the impedance values from the ANN output also match the measured one very well. Above 200 MHz (such frequency value is usually much beyond the frequency limit for which the PDN and decap placement needs to be optimized), the probably induced numerical damping is evident.

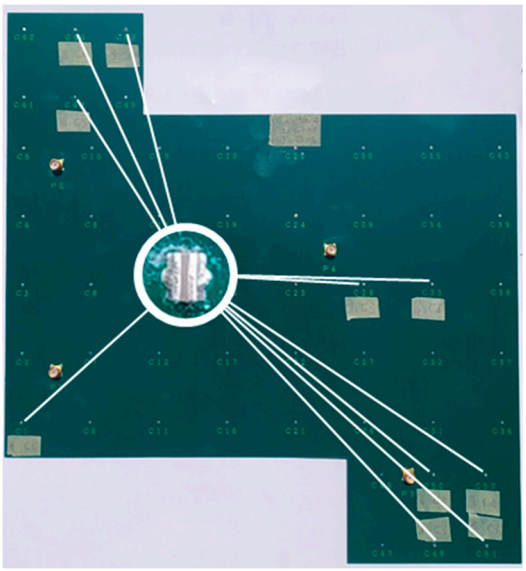
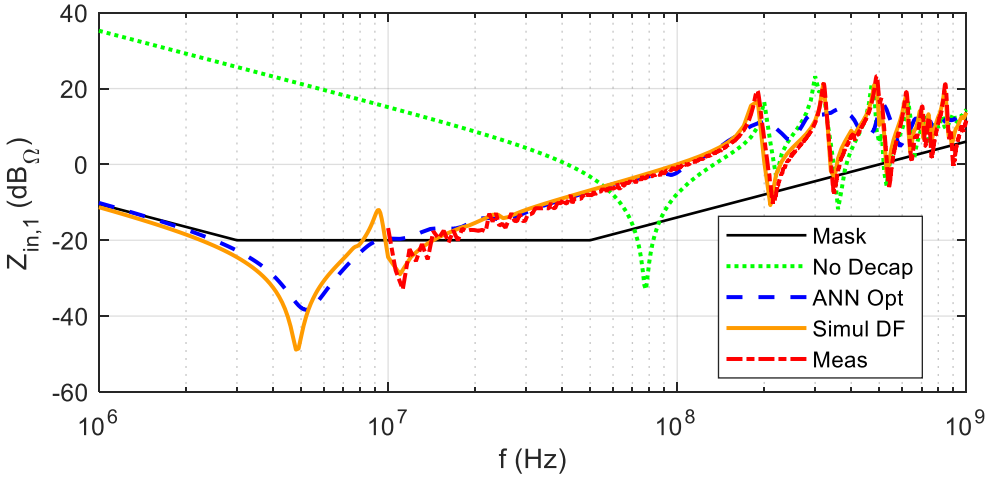
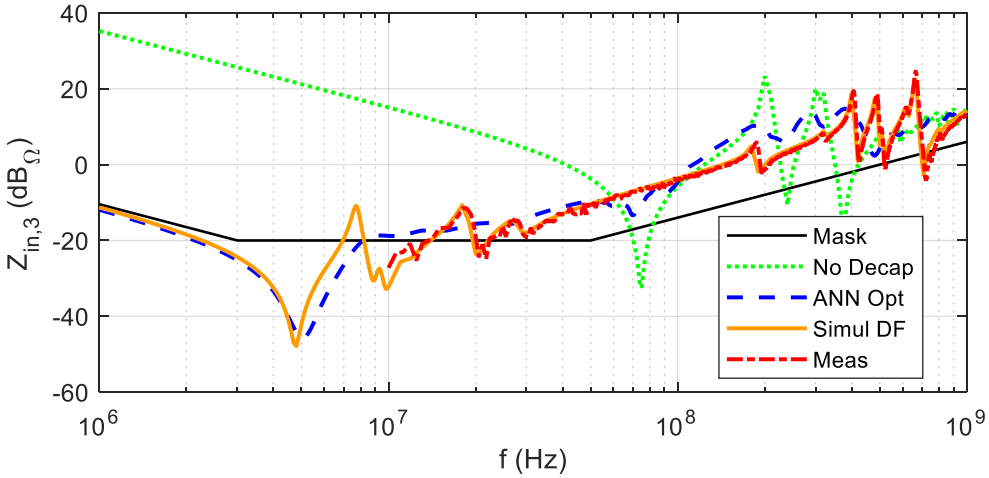


Figure 13. Assembled PCB and identification of the 10 decap locations.



(**a**)



(**b**)

Figure 14. Input impedances at ports 1 (**a**) and 3 (**b**). Comparisons among the impedances measured, simulated by DF, and predicted by the ANN.

5. Conclusions

An alternative way to evaluate the input impedance of a PDN is developed in this paper based on an ANN. The obtained ANN, once accurately trained, can be efficiently used into an optimization algorithm for finding the best decap placement from a given target impedance. The ANN, although its accuracy is lower than a full-wave simulator, is a much quicker tool that PDN commercial software. The output provided by the GA–ANN based process is consistent with the main principles for a PDN design, and with the results obtained by using the GA in combination with a commercial tool. The impedance predicted by the ANN agrees with those obtained by the DF simulator and by hardware measurements.

The target of this work was to quickly obtain the first solution that fulfils the mask limit (considered the optimal) without the need to look to better solutions. The statement is supported by the fact that multiple optimizations based on the GA lead to similar results in terms of cost functions, thus confirming that the optimum solution is not unique. This can be surely tolerated as long as other constraints are not applied in the calculation of the cost function; i.e., the weight associated to the cost of a decaps, or the weight associated to specific (preferred) locations. The examples in Figures 10 and 12 confirm the validity of the above assumption, with the impedance evaluated by the developed ANN; in such cases, the cost function in Figure 10b and in Figure 12b are very similar based on the same input, even though the obtained decap configuration in Figures 10a and 12a are slightly different. A similar comparison can be seen in Figure 6 in [19], where the cost functions up to the sixth iteration are almost overlapped.

The obtained results open at least two research focuses that will be considered next: the investigation of unwanted numerical effects on the frequency spectrum of the impedance around some specific resonance or anti-resonance frequencies, and the implementation of multiple executions of the GA in order to improve the identification of the optimal solution by means of a robust statistical analysis.

Author Contributions: Conceptualization, A.O. and M.B.; methodology, A.O.; software, C.O., R.C. and M.B.; validation, F.d.P.; writing—original draft preparation, A.O., R.C., and F.d.P.; writing—review and editing, F.d.P. and A.O. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Google, 2017 Google Faculty Research Award.

Acknowledgments: The authors would like to thank Zhiping Yang at Google for his valuable support, the useful discussions and the insightful comments for the development of this work and ZUKEN GmbH for having supported the project with its software.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yang, G.; Chen, K.; Yu, Z.; Zhang, Y.; Zhou, F.; He, J. An Inversion Method for Evaluating Lightning Current Waveform Based on Time Series Neural Network. *IEEE Trans. Electromagn. Compat.* **2017**, *59*, 887–893. [CrossRef]
2. da Arantes, J.; da Arantes, M.; Missaglia, A.B.; Simoes, E.D.V.; Toledo, C.F.M. Evaluating Hardware Platforms and Path Re-planning Strategies for the UAV Emergency Landing Problem. In Proceedings of the 2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI), Boston, MA, USA, 6–8 November 2017; pp. 937–944. [CrossRef]
3. Ku, C.K.; Goay, C.H.; Ahmad, N.S.; Goh, P. Jitter Decomposition of High-Speed Data Signals From Jitter Histograms With a Pole–Residue Representation Using Multilayer Perceptron Neural Networks. *IEEE Trans. Electromagn. Compat.* **2019**, 1–11. [CrossRef]
4. Piersanti, S.; Orlandi, A.; de Paulis, F. Electromagnetic Absorbing Materials Design by Optimization Using a Machine Learning Approach. *IEEE Trans. Electromagn. Compat.* **2018**, 1–8. [CrossRef]
5. Lu, T.; Wu, K. Machine learning methods in high-speed channel modeling. In Proceedings of the DesignCon 2019, Santa Clara, CA, USA, 29–31 January 2019.
6. Lu, T.; Sun, J.; Wu, K.; Yang, Z. High-Speed Channel Modeling with Machine Learning Methods for Signal Integrity Analysis. *IEEE Trans. Electromagn. Compat.* **2018**, *60*, 1957–1964. [CrossRef]

7. Xu, J.; Bai, S.; Nalla, K.; Sapozhnikov, M.; Drewniak, J.L.; Hwang, C.; Fan, J. Power Delivery Network Optimization Approach using an Innovative Hybrid Target Impedance. In Proceedings of the 2019 IEEE International Symposium on Electromagnetic Compatibility, Signal & Power Integrity (EMC+SIPI), New Orleans, LA, USA, 22–26 July 2019; pp. 211–216.
8. Oh, D.; Razmadze, A.; Chandrasekar, K. Power integrity analysis for core logic blocks. In Proceedings of the 2013 IEEE 22nd Conference on Electrical Performance of Electronic Packaging and Systems, San Jose, CA, USA, 27–30 October 2013; pp. 79–82.
9. Kim, J.; Wu, S.; Wang, H.; Takita, Y.; Takeuchi, H.; Araki, K.; Feng, G.; Fan, J. Improved target impedance and IC transient current measurement for power distribution network design. In Proceedings of the IEEE International Symposium on Electromagnetic Compatibility, Fort Lauderdale, FL, USA, 25–30 July 2010; pp. 445–450.
10. Armstrong, C. Debug and analysis considerations for optimizing signal integrity in your internet of things design. In Proceedings of the 2017 IEEE International Symposium on Electromagnetic Compatibility & Signal/Power Integrity (EMCSI), Washington, DC, USA, 7–11 August 2017; pp. 20–24.
11. De Paulis, F.; Zhao, B.; Piersanti, S.; Cho, J.; Cecchetti, R.; Achkir, B.; Orlandi AFan, J. Impact of chip and interposer PDN to eye diagram in high speed channels. In Proceedings of the 2018 IEEE 22nd Workshop on Signal and Power Integrity (SPI), Brest, France, 22–25 May 2018; pp. 1–4.
12. Piersanti, S.; de Paulis, F.; Olivieri, C.; Orlandi, A. Decoupling Capacitors Placement for a Multichip PDN by a Nature-Inspired Algorithm. *IEEE Trans. Electromagn. Compat.* **2018**, *60*, 1678–1685. [CrossRef]
13. Swaminathan, M.; Han, K.J. *Design and Modeling for 3D ICs and Interposers*; World Scientific: Singapore, 2014.
14. Smith, L.D.; Bogatin, E. *Principles of Power Integrity for PDN Design*; Prentice-Hall: New York, NY, USA, 2016.
15. Zamek, I.; Boyle, P.; Li, Z.; Sun, S.; Chen, X.; Chandra, S.; Li, T. Modeling FPGA current waveform and spectrum and PDN noise estimation. In Proceedings of the DesignCon 2008, Santa Clara, CA, USA, 22 February 2008.
16. Erdin, I.; Achar, R. Multi-pin optimization method for placement of decoupling capacitors using genetic algorithm. *IEEE Trans. Electromagn. Compat.* **2018**, *60*, 1662–1669. [CrossRef]
17. Erdin, I.; Achar, R. Multi-Objective Optimization of Decoupling Capacitors for Placement and Component Value. *IEEE Trans. Compon. Packag. Manuf. Technol.* **2019**, *9*, 1976–1983. [CrossRef]
18. Song, E.; Koo, J.; Pak, J.S.; Kim, J. Through-Silicon-Via-Based Decoupling Capacitor Stacked Chip in 3-D-ICs. *IEEE Trans. Compon. Packag. Manuf. Technol.* **2013**, *9*, 1467–1480. [CrossRef]
19. de Paulis, F.; Cecchetti, R.; Olivieri, C.; Piersanti, S.; Orlandi, A.; Buecker, M. Efficient Iterative Process Based on an Improved Genetic Algorithm for Decoupling Capacitor Placement at Board Level. *Electronics* **2019**, *8*, 1219. [CrossRef]
20. Kahng, S. GA-optimized decoupling capacitors damping the rectangular power-bus' cavity-mode resonances. *IEEE Microw. Wirel. Compon. Lett.* **2006**, *16*, 375–377. [CrossRef]
21. Shringarpure, K.; Zhao, B.; Wei, L.; Archambeault, B.; Ruehli, A.; Cracraft, M.; Cocchini, M.; Wheeler, E.; Fan, J.; Drewniak, J. On finding the optimal number of decoupling capacitors by minimizing the equivalent inductance of the PCB PDN. In Proceedings of the 2014 IEEE International Symposium on Electromagnetic Compatibility (EMC), Raleigh, NC, USA, 4–8 August 2014; pp. 218–223.
22. Kadlec, P.; Marek, M.; Štumpf, M.; Šeděnka, V. PCB Decoupling Optimization With Variable Number of Capacitors. *IEEE Trans. Electromagn. Compat.* **2019**, *61*, 1841–1848. [CrossRef]
23. Su, H.; Sapatnekar, S.S.; Nassif, S.R. Optimal decoupling capacitor sizing and placement for standard-cell layout designs. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **2003**, *22*, 428–436. [CrossRef]
24. Wang, X.; Cai, Y.; Zhou, Q.; Tan, S.X.; Eguia, T. Decoupling capacitance efficient placement for reducing transient power supply noise. In Proceedings of the 2009 International Conference on Computer-Aided Design, San Jose, CA, USA, 2–5 November 2009; pp. 745–751.
25. Zuken. Design Force User Manual. Available online: https://www.zuken.com/it/products/pcb-design/cr-8000/products/design-force (accessed on 2 September 2019).
26. Piersanti, S.; Cecchetti, R.; Olivieri, C.; de Paulis, F.; Orlandi, A.; Buecker, M. Decoupling Capacitors Placement at Board Level Adopting a Nature-Inspired Algorithm. *Electronics* **2019**, *8*, 737. [CrossRef]
27. Haupt, R.L.; Werner, D.H. *Genetic Algorithms in Electromagnetics*; John Wiley & Sons: New York, NY, USA, 2007.
28. Bishop, C.M.; Roach, C.M. Fast curve fitting using neural networks. *Rev. Sci. Instrum.* **1992**, *63*, 4450–4456. [CrossRef]

29. Matlab Manual: Improve Shallow Neural Network Generalization and Avoid Overfitting MATLAB & Simulink. Available online: https://it.mathworks.com/help/deeplearning/ug/improve-neural-network-generalization-and-avoid-overfitting.html (accessed on 31 July 2020).
30. TDK. Available online: https://product.tdk.com/en/search/capacitor/ceramic/mlcc/characteristic/ (accessed on 4 February 2019).
31. MacKay, D.J.C. Bayesian interpolation. *Neural Comput.* **1992**, *4*, 415–447. [CrossRef]
32. Demuth, H.B.; Beale, M.H.; De Jess, O.; Hagan, M.T. *Beale: Neural Network Design*; Martin Hagan: Stillwater, OK, USA, 2014.
33. Ortega, J.G.; Janer, C.L.; Quero, J.M.; Franquelo, L.G.; Pinilla, J.; Serrano, J. Analog to digital and digital to analog conversion based on stochastic logic. In Proceedings of the IECON '95-21st Annual Conference on IEEE Industrial Electronics, Orlando, FL, USA, 6–10 November 1995; Volume 2, pp. 995–999.
34. de Paulis, F.; Cecchetti, R.; Olivieri, C.; Buecker, M. Genetic Algorithm PDN Optimization based on Minimum Number of Decoupling Capacitors Applied to Arbitrary Target Impedance. In Proceedings of the IEEE International Symposium on Electromagnetic Compatibility and Signal & Power Integrity, 3–28 August 2020. www.ieee.com.