Augmented Genetic Algorithm v3 for Multi-Objective PDN Decap Optimization

Haran Manoharan[#1],Jack Juang[#2], Hanfeng Wang[*3], Jingnan Pan[*4], Xu Gao[*5], Kelvin Qiu [*6], and Chulsoon Hwang[#7]

[#]*EMC Laboratory,* Missouri University of Science and Technology, Rolla, Mo, USA
[1]hm6h6, [2]jjryb, [7]hwangc@mst.edu
[*]*Google Inc, Mountain View, CA, USA*
[3]hanfengw, [4]jingnan,[5]xugaon, [6]kqiu@google.com

***Abstract*—This paper proposes a novel approach for generating the initial population, termed disproportionate initialization, to enhance the genetic algorithm's efficiency in optimizing the number of decaps. This method is specifically designed to accelerate convergence, improving the algorithm's ability to find optimal solutions. The algorithm is improved to simultaneously reduce the number of decoupling capacitors and achieve the lowest cost, aiming to identify the global minima for comprehensive PDN optimization.**

Keywords—Multi-Objective Genetic Algorithm, disproportionate initial population, population diversity, decap optimization

I. Introduction

Recent research in PDN optimization has primarily focused on reducing the number of decoupling capacitors using machine learning techniques. Optimizing the selection of decoupling capacitors (decaps) in a power distribution network (PDN) is pivotal for reducing impedance and cost. Numerous research has utilized machine learning methods to optimize both the placement and values of decaps.

In work by [1], a transformer network-based deep reinforcement learning agent was employed to optimize decap placement; however, this approach is constrained to using single decap, making it specifically suitable for single decap placement issues. In practical scenarios, capacitor libraries encompass numerous decaps that necessitate optimization. Another attempt in [2] utilized a deep reinforcement learning approach for optimizing decap placement and value. However, this algorithm had issues with scalability and generalizing to different boards. [3] proposed an advantage actor-critic reinforcement learning-based method, offering a broader action space. Nonetheless, scalability and reusability remain challenges for this method.

The exploration of evolutionary computation techniques led [4] to propose a genetic algorithm for decap optimization. Although computationally expensive, employing a commercial tool for impedance calculations in each population, this method laid the foundation for further enhancements. Subsequently, [5] introduced a modified version of conventional genetic algorithms, termed gene-suppressed GA, aiming for faster convergence. [6] presented a genetic algorithm with an augmented initial population, incorporating new genetic operators. While exhibiting commendable performance in terms of solution quality and time cost, this algorithm faced challenges related to convergence, occasionally becoming stuck in local minima. The algorithm proposed in [7] is an improved version of the algorithm in [6] with reinforcement learning (RL) for tuning the mutation probability of the genetic algorithm. The algorithm, with the addition of the RL agent, was able to converge faster and was able to find a better decap solution than the algorithm in [6]. The existing approaches collectively contribute to the evolving landscape of PDN decap optimization, each with its distinctive advantages and limitations. However, in many practical industrial scenarios, the optimization requirements extend beyond minimizing the capacitor count.

In this work, the algorithm proposed in [7] is improved further with the extension of finding a solution with fewer decaps and minimal cost. In addition to this, the initial population of the algorithm is improved with disproportionate decap solutions in the initial population to allow for faster convergence and find a near-global minimum.

II. Previous Work

A. Augmented GA v1

A GA is a population-based optimization algorithm used to solve combinatorial problems. The initial population is important for the algorithm to converge faster to the near global minimum.

Hence, in augmented GA v1, an augmented initial population was proposed. By finding the decap weights, the best proportion of decaps from the decap library needed for the board is generated. The decap solution is encoded as a vector of real numbers where the index corresponds to decap ports, and values in each index correspond to the decap type in the decap library.

For solutions that satisfy the target impedance, the fitness is given by (1), and for solutions that do not satisfy the target impedance, the fitness is given by (2) [6].

$$\text{fitness} = -(\text{total \# of ports - \# of ports used}) + 1 \tag{1}$$

$$\text{fitness}=\max\left(\frac{\text{solution\_z(f)-target\_z(f)}}{\text{target\_z(f)}}\right) \tag{2}$$

Apart from the initial population, new mutation operators were introduced, and crossover operation was removed. Helper functions for faster convergence were also used. Compared to conventional GA, the algorithm showed improved solution quality and took less computation time. Fig. 1 shows the flowchart of the augmented GA v1.

Copyright © 2024 IEICE

Authorized licensed use limited to: Missouri University of Science and Technology. Downloaded on January 16,2025 at 18:54:50 UTC from IEEE Xplore. Restrictions apply.

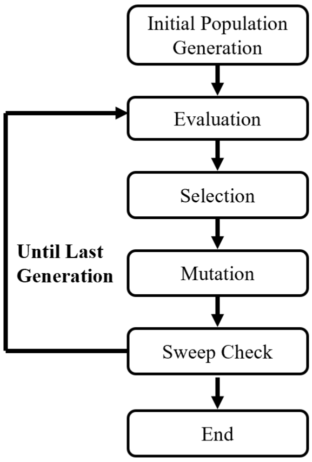

Fig. 1 Augmented GA v1 flowchart.

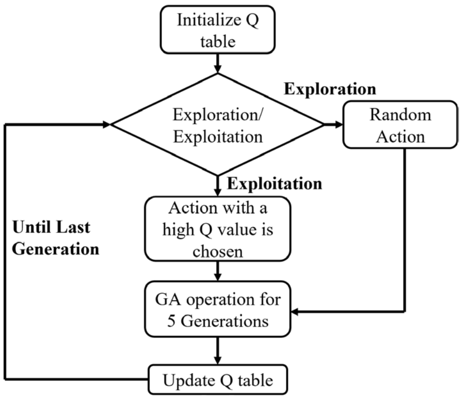

Fig. 2 Selection of mutation probability by RL agent

B. Augmented GA v2

Mutation probability is a hyperparameter of GA and is predefined before the start of the algorithm. It was found that proper selection of mutation probability will make the algorithm find a better solution. Since this is different for different board cases, an intelligent way of tuning the mutation probability was proposed in v2[7]. Q learning-based reinforcement learning was used to tune the mutation probability. Fig. 2 shows the flowchart of the selection of mutation probability by the RL agent. The algorithm starts with a predefined mutation probability, and for every 5 generations until the exploration stage, the algorithm chooses a random mutation probability and updates its Q table.

The reward for the agent was defined as if there was an improvement in GA's performance in terms of finding a better solution, and then a positive reward is awarded or vice-versa. Then, in the exploitation phase, the agent takes the action that tries to maximize the reward. In this way, the agent assists the GA in finding an optimal solution or near-optimal solution instead of getting stuck in the local minimum.

The performance results were compared with v1 and conventional GA for 30 cases, and in all cases, algorithm v2 outperformed in terms of solution quality and computation time.[7]

III. Augmented Genetic Algorithm V3

In both v1 and v2, the initial population is filled with decap solutions where all the ports are filled. If the population size is 20, then all the ports in these 20 decap solutions will have their ports filled. This leads to a case where there is no diversity in the initial population. Population diversity is really needed to improve the performance of the genetic algorithm. Hence, in this work, v3 disproportionate initial population is proposed to maintain the population diversity.

A. Disproportionate Initial Population

In this proposed approach, the initial population is divided into four categories. These four categories include 100% filled, 80% filled, 60% filled, and 40% filled decap ports solution in the initial population. The ports are filled based on the port priority as in v1[6]. If the population size is 20, then each category will have 4 solutions. Instead of starting with an initial population of all decap ports by the proposed method, there is a diverse initial population. This diverse initial population helps the convergence of the algorithm faster and, in some cases, finds better solutions. There are often cases that the optimized solution requires much fewer decaps than the available decap ports. For example, with 50-board and loose target impedance, the global minimum is 25 decaps. With this new initial population, the 60% filled decap ports solution (30 ports in this case) will satisfy the target impedance, and the algorithm now starts with a solution closer to the global minimum eventually, as the algorithm iterates, it will find the global minimum or near global minimum.

To validate the performance of the proposed new initial population-based GA, 3 cases each for 75,100 and 150 decap ports were considered. These 3 cases included tight, loose, and very loose target impedance. The performance was tested against v2 in terms of solution quality and computation time. All cases were run on a Linux server that had 12 cpu cores. The comparison results are shown in Table I. For the 75 board in all 3 cases, both the v2 and v3 converge to the same solution, but v3 is faster than v2. For the 100 and 150 board cases, the v3 outperforms the v2 in terms of solution quality, finding a better solution. The same decap library used in [6] is used here too.

TABLE I. Comparison of solution quality and time cost

Example #	Minimum # of capacitors needed		Time taken	
	v2	*v3*	*v2*	*v3*
75 decap ports case #1	27	27	~19 mins	~12 mins
75 decap ports case #2	21	21	~15 mins	~10 mins
75 decap ports case #3	13	13	~10 mins	~6 mins
100 decap ports case #1	34	34	~27 mins	~20 mins
100 decap ports case #2	29	28	~25 mins	~16 mins
100 decap ports case #3	25	24	~20 mins	~12 mins
150 decap ports case #1	45	44	~48 mins	~34 mins
150 decap ports case #2	34	34	~42 mins	~30 mins
150 decap ports case #3	16	15	~36 mins	~25 mins

On an overall case with a diverse initial population, convergence is faster, and it allows the algorithm to explore and find a near-optimal solution. Fig. 3 shows the convergence plot comparison for the three 75 decap port cases shown in

Copyright © 2024 IEICE

Authorized licensed use limited to: Missouri University of Science and Technology. Downloaded on January 16,2025 at 18:54:50 UTC from IEEE Xplore. Restrictions apply.

Table I. In the plot, blue, pink, and black show results for v2, and red, purple, and brown show results for v3. Comparing the blue and red curves, it shows that v3 finds the solution with 27 decaps in the 5th generation while v2 takes the 51st generation to find the same solution with 27 decaps. The proposed initial population method made the GA converge faster to the minimum solution.

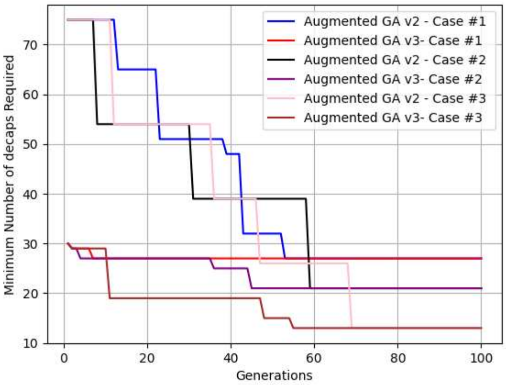

Fig. 3 Convergence comparison plot of three 75 decap port cases

Comparing the proposed approach to the full search, an exhaustive search was carried out to check if the algorithm could find the global minimum possible. For a 150-board case and 10 different decaps in the library, the number of combinations is around two thousand billion (2.1e15) samples. Carrying out this number of combinations is beyond the capability. Hence, a small board case with 50 decap ports and 4 decaps in the decap library is considered. This will lead to a total of 292825 samples. The comparison is shown in Table II. In the 50 decap ports case, the proposed algorithm is able to find the near-global minimum solution. However, the time taken for an exhaustive search is almost 2 days compared to the 15 minutes taken by the proposed approach.

TABLE II. COMPARISON WITH EXHAUSTIVE SEARCH

Example #	**Minimum # of capacitors needed**		**Time Taken**	
	v3	***Exhaustive search***	***v3***	***Exhaustive search***
50 decap ports case #1	26	25	~14 mins	~2 days

B. Multi-Objective Optimization

In this proposed version, the GA is modified to handle multi-objective optimization. To handle this task, the reward function is modified to a normalized version. Since the scaling differs for the number of decaps and the total cost, the normalized reward function is used to make the GA unbiased to one objective. The modified reward function is given by (3)

$$\text{reward} = \text{reward1}+\text{reward2} \tag{3}$$

$$\text{reward1} = \frac{\text{-(total \# of ports - \# of ports used)}}{\text{total \# of ports}} \tag{4}$$

$$\text{reward2}= \frac{\text{total cost of decaps in the solution}}{\text{total cost possible}} \tag{5}$$

Table III gives the results for the multi-objective decap optimization. Test cases were carried out for 25 and 50 board cases. The results show with the new reward function, the algorithm can handle both objectives (a smaller number of decaps and minimal cost).

TABLE III. MULTI-OBJECTIVE OPTIMIZATION RESULTS

Example #	***Minimum # of capacitors needed***	***Minimum Cost***
25 decap ports	10	$1
50 decap ports	27	$2.7

IV. CONCLUSION

Population diversity is maintained by having a disproportionate initial population that aides in faster convergence of the algorithm. In addition to that, the reward function is modified to support multi-objective optimization of both the number of decaps and cost is implemented. In comparison with v2, the proposed algorithm v3 shows improved solution quality with less computation time. The solution obtained is compared with the exhaustive search, and the proposed algorithm finds the near global minimum. In the future, the algorithm performance will be improved, and a minimum decap area objective will be added in the multi-objective optimization.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation (NSF) under Grant. IIP-1916535.

REFERENCES

[1] H. Park et al., “Deep Reinforcement Learning-Based Optimal Decoupling Capacitor Design Method for Silicon Interposer-Based 2.5-D/3-d ICs,” IEEE Trans. Compon. Packaging Manuf. Technol., vol. 10, no. 3, pp. 467-478, March 2020, doi: 10.1109/TCPMT.2020.2972019.

[2] L. Zhang, W. Huang, J. Juang, H. Lin, B. -C. Tseng and C. Hwang, “An Enhanced Deep Reinforcement Learning Algorithm for Decoupling Capacitor Selection in Power Distribution Network Design,” 2020 IEEE International Symposium on Electromagnetic Compatibility & Signal/Power Integrity (EMCSI), 2020, pp. 245-250, doi: 10.1109/EMCSI38923.2020.9191512.

[3] S. Han, O. W. Bhatti and M. Swaminathan, "Reinforcement Learning for the Optimization of Decoupling Capacitors in Power Delivery Networks," 2021 IEEE International Joint EMC/SI/PI and EMC Europe Symposium, 2021, pp. 544-548, doi: 10.1109/EMC/SI/PI/EMCEurope52599.2021.9559342

[4] F. de Paulis, R. Cecchetti, C. Olivieri and M. Buecker, "Genetic Algorithm PDN Optimization based on Minimum Number of Decoupling Capacitors Applied to Arbitrary Target Impedance," 2020 IEEE International Symposium on Electromagnetic Compatibility & Signal/Power Integrity (EMCSI), 2020, pp. 428-433, doi: 10.1109/EMCSI38923.2020.9191458.

[5] J. Juang, L. Zhang, Z. Kiguradze, B. Pu, S. Jin and C. Hwang, "A Modified Genetic Algorithm for the Selection of Decoupling Capacitors in PDN Design," 2021 IEEE International Joint EMC/SI/PI and EMC Europe Symposium, 2021, pp. 712-717, doi: 10.1109/EMC/SI/PI/EMCEurope52599.2021.9559292.

[6] J. Juang *et al.*, “Augmented Genetic Algorithm for Decoupling Capacitor Optimization in PDN Design Through Improved Population Generation," submitted to IEEE Trans. on Signal and Power Integrity.

[7] H. Manoharan, J.Juang, J.Pan, H.Wang, K.Qiu and C.Hwang, "Augmented Genetic Algorithm v2 with Reinforcement Learning for PDN Decap Optimization," 2023 IEEE Symposium on

Copyright © 2024 IEICE

Authorized licensed use limited to: Missouri University of Science and Technology. Downloaded on January 16,2025 at 18:54:50 UTC from IEEE Xplore. Restrictions apply.

Electromagnetic Compatibility & Signal/Power Integrity (EMC+SIPI), Grand Rapids, MI, USA, 2023, pp. 255-258, doi: 10.1109/EMCSIPI50001.2023.10241752

Copyright © 2024 IEICE

Authorized licensed use limited to: Missouri University of Science and Technology. Downloaded on January 16,2025 at 18:54:50 UTC from IEEE Xplore. Restrictions apply.