Optimized DC-DC Boost Converter with Inverted Gamma Filter Based on Genetic Algorithm

SalahAldeen AlZoubi
Department of Electrical Engineering
Princess Sumaya University for Technology
Amman, Jordan
sal20208181@std.psut.edu.jo

Awos Kanan
Department of Computer Engineering
Princess Sumaya University for Technology
Amman, Jordan
a.kanan@psut.edu.jo

Abstract— **In this paper, a genetic algorithm is used to optimize of a Direct Current to Direct Current boost converter with inverted-gamma filter. The objective is to reduce the electro-magnetic interference (EMI) produced from the non-linear elements by minimizing the control-to-output transfer function. The optimization was done using MATLAB/Global optimization toolbox for 3 different initial populations, each population was optimized using 5 different genetic algorithm operations, for a total of 16 options and 48 runs. Experimental results show that the proposed approach achieved an average fitness function value of (-234.87), which is 5.41% better fitness function value compared to a previous related work.**

Keywords—Boost Converters, EMI, Optimization, Genetic Algorithm.

I. Introduction

Direct Current to Direct Current (DC-DC) converters play a significant role in nowadays technology. They can be nearly found in every DC operating device, such as electric vehicles, electric power generation (wind and photovoltaic), and mobile phones. DC-DC converters are particularly useful in changing the DC voltage from a level to another, through a circuit that consists of resistors, inductors, capacitors, diodes, and switching devices [1]. DC-DC converters also generate current and voltage related interference at the input and the output of the converters, resulting in noise injection to the power system, and a disturbed operation of communication and control systems [2].

To avoid such interferences, active and passive filters are used. Active filter design is complicated due to the complexity involved in its operation. Passive filters, on the other hand, have been traditionally used for the mitigation of harmonic distortion [2]. The work proposed in this study is for optimizing a DC-DC boost converter with a passive inverted-gamma filter. The inverted-gamma filter is a type of LC filters that varies the input impedance of the converter, resulting in a difficult optimization of the EMI filter design. The optimized parameters in this study are the inverted-gamma filter inductor and capacitor, the boost converter inductor and capacitor, and the switching frequency using metaheuristic optimization. Metaheuristic algorithms are high-level algorithms designed to find optimal or near-optimal solutions for an optimization problem, especially when we do not have enough knowledge about the optimization problem itself [3].

Several approaches have been proposed in the literature to optimize different parameters of DC-DC converters such as electromagnetic interference (EMI) [4], inductor value [5], efficiency [2], reliability [6], and controller tuning [7]. Different types of metaheuristics algorithms such as genetic algorithm, cuckoo algorithm, and earthquake algorithm have been used to find optimal values for the selected parameters. Reference [4] used the genetic algorithm because it is less dependent on the initial population. Reference [5] used the earthquake algorithm due to its ability to provide both wide and fine searching paths depending on the given velocities. The authors of [2] and [6] used the genetic algorithm due its capability of crossing local minima. Reference [7] used the cuckoo algorithm due to its fast convergence rate and global search abilities. The genetic algorithm is used in this paper due to its effectiveness in large search space, continuous variables, and parameter tuning optimization problems.

In this paper, a genetic algorithm is proposed to enhance the performance of DC-DC boost converters with inverted-gamma filters by minimizing the EMI generated by non-linear elements. A Genetic algorithm is a search-based optimization technique that simulates the process of natural selection, which applies the principle of survival of the fittest. It mimics the evolution process on the problem to be optimized by evolving a set of populations iteratively to reach a better solution [8].

The remaining of this paper is organized as follows. The optimization problem is formulated in Section II. Section III presents the proposed methodology. Experimental results are presented and discussed in Section IV. Section V presents the evaluation and a mathematical proof of the results. Finally, Section VI concludes the paper.

II. Problem Formulation

A. System Modeling

DC-DC boost converters consist of linear and nonlinear elements. The nonlinear elements will complicate the calculation of the DC operating point. Hence, the small signal analysis approach is used to approximate the behaviour of a nonlinear power electronics system, with a linear time-invariant (LTI) model that is valid around an operating point of interest. Small-signal analysis is an enabling step to apply classic control theory to power electronics systems, which requires an LTI representation such as a transfer function or a state-space model of the system. Linearizing techniques are proposed in reference [9]. Figure 1 shows the circuit diagram of the boost converter with inverted gamma filter. Figure 2 shows the boost converter after linearization.

The small signal model is used to derive the control-to-output transfer function (Gvd), which will be used to

2023 14th International Conference on Information and Communication Systems (ICICS) | 979-8-3503-0786-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICICS60529.2023.10330452

determine the converter behaviour at a specific frequency. Equation (1) represents the control to output transfer function of the boost converter with inverted gamma filter [4].

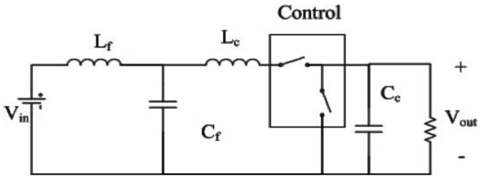

Fig. 1 Boost Converter with Inverted Gamma Filter Circuit Diagram [4].

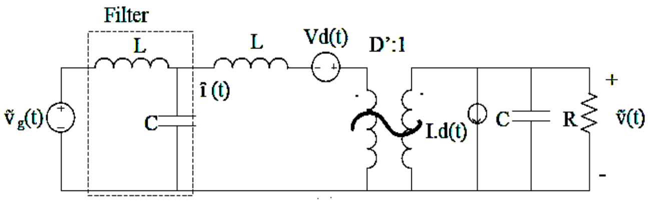

Fig. 2 The Converter after Linearization [4].

Where:
L_c : *Boost inductor*
L_f : *Filter inductor*
C_c: *Boost capacitor*
C_f: *Filter capacitor*
D : *Duty Cycle*
D': 1: *Transformer turns ration*
V: *The voltage*
I: *The current*

$$G_{vd}(s) = \frac{V_o}{d_p}\frac{sw_1+w_2}{s^2u_1+su_2+u_3}\frac{s^4x_1+s^3x_2+s^2x_3+sx_4+x_5}{s^3y_1+s^2y_2+sy_3+y_4} \quad (1)$$

Where:

$w_1 = -L_c;\ w_2 = d_pR;\ u_1 = L_cC_cR;\ u_2 = L_c;\ u_3 = d_p^2R;$
$x_1 = L_fC_fL_c^2d_pRC_c$
$x_2 = [L_f + L_cd_pR - d_p^2RL_fC_f]L_cC_c + L_c^2L_fC_fd_pR$
$x_3 = [L_f + L_cd_pR - d_p^2RL_fC_f]L_c + L_fC_fL_cd_p^3R^2$
$x_4 = [L_f + L_cd_pR - d_p^2RL_fC_f]d_p^2R - d_p^2RL_c$
$x_5 = -d_p^4R^2\ ;\ y_1 = L_cC_cL_fC_f;\ y_2 = L_cC_c + L_cL_fC_f;$
$y_3 = L_c + L_f + L_fC_fd_p^2R\ ;\ y_4 = d_p^2R$
Vo: *Output voltage*
$L_c(X[1]) = Boost\ inductor$
$L_f(X[2]) = Filter\ inductor$
$C_c(X[3]) = Boost\ capacitor$
$C_f(X[4]) = Filter\ capacitor$
$d_p = 1 - the\ duty\ cycle$
$R = Load\ resistor$

B. Encoding

The encoding method used in this work is value encoding, where a chromosome represents a solution to the optimization problem, each gene represents an optimization variable, and the allele will be the value of the variable. The chromosome consists of five genes that represent the five optimization variables: L_c, L_f, C_c, C_f, and f. Figure 3 shows the chromosome and the genes of this problem.

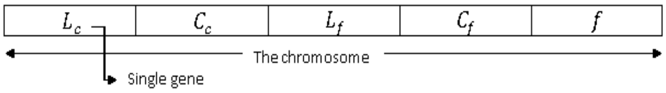

Figure 3. Chromosome and Genes of the Problem

C. Fitness Function & Constraints

The fitness function, $f(x)$, is shown in Equation (2). Equations (4) -(6) show the constraints used to ensure proper working of the converter.

$$f(x) = -20\log|G_{vd}(j\omega)| \quad (2)$$

$$L_c - \frac{D.T.V_{in}}{2.\Delta i} = 0 \quad (3)$$

$$C_c - \frac{D.T.V_o}{2.\Delta v.R} = 0 \quad (4)$$

$$10^{-9} \leq X[1,2,3,4] \leq 10^{-2} \quad (5)$$

$$10^3 \leq f \leq 15x10^4 \quad (6)$$

Where X[1,2,3,4] are the values of inductors and capacitors.

III. Methodology

The optimization process was done using MATLAB/Global Optimization Toolbox. Different random seeds were used to start at random initial populations. The same seed has been used to ensure fair comparison between different options using the same initial population instance. Changing the seed allows monitoring the behaviour of the same options over different initial population instances. The selected options from selection function are roulette selection, tournament selection, and remainder selection. The selected options from crossover function are scattered crossover, heuristic crossover, single point crossover, and double point crossover. The selected options from mutation function are Gaussian mutation, uniform mutation, and power mutation [10]. The selected numbers for elite members are one, three, and five. The selected numbers for population size are five, ten, and twenty for three different seeds.

The GA will start by creating random initial population, then the individuals will be selected depending on the fitness value and a randomized selection process depending on a probability, these selected individuals will create a new population. Some individuals in the new population will change due to crossover and mutation, which will produce new individuals, these operations continue until a stopping criterion is met. Figure 4 shows the flow chart of the genetic algorithm.

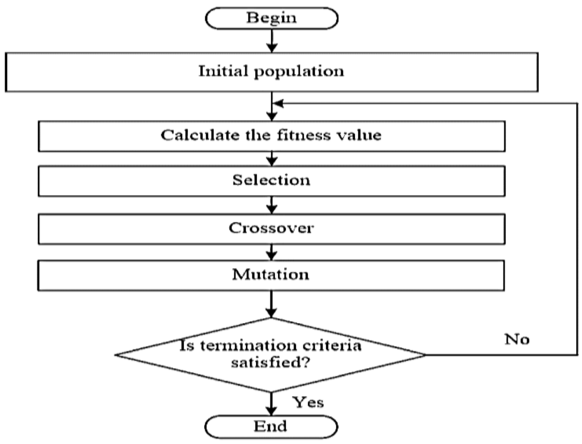

Figure 4. Flow chart of the genetic algorithm.

IV. Results

In this section, the options of the proposed genetic algorithm are fine-tuned to help reach optimal values of the optimization variables. Subsections A, B, C, D, and E represent the selection, crossover, mutation, number of elite members, and the number of individuals respectively for three different seeds. Seed 1 starting parameters are $L_C = 1x10^{-9}H$, $L_f = 1x10^{-9}H$, $C_c = 1x10^{-9}F$, $C_f = 1x10^{-9}F$ & $f = 1x10^{3}Hz$. Seed 2 starting parameters are $L_C = 5.39x10^{-7}H, L_f = 9.8514x10^{-4}\ H$, $C_c = 0.0057\ F$, $C_f = 8.004x10^{-4}\ F$ & $f =$ 29381 Hz. Seed 3 starting parameters are $L_C = 5.3208\ x\ 10^{-6}\ H$, $L_f = 7.2352\ x\ 10^{-4}\ H$, $C_c = 4.6078\ x\ 10^{-5}\ F$, $C_f = 0.0099\ F$, $f = 11346\ Hz$.

A. Selection Operation

Selection operation is the operation of choosing the parents of the next generation; it has different options such as roulette, tournament, and remainder selection. Roulette selection chooses parents by simulating a roulette wheel; the area of each section is proportional to the individual fitness value. Tournament selection initiates random parents then chooses the parents with best fitness value. Remainder selection assigns parents deterministically from the integer part of each individual's scaled value and then uses roulette selection on the remaining fractional part. Figures (5)-(7) show the behaviour of the different selection options in the three different seeds.

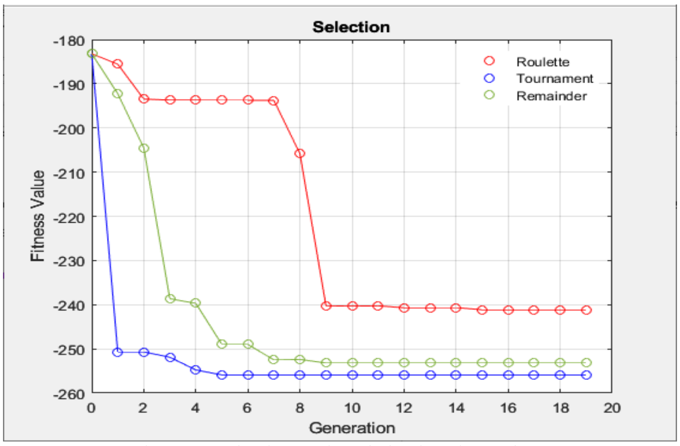

Figure 5. Selection options behaviour in seed 1

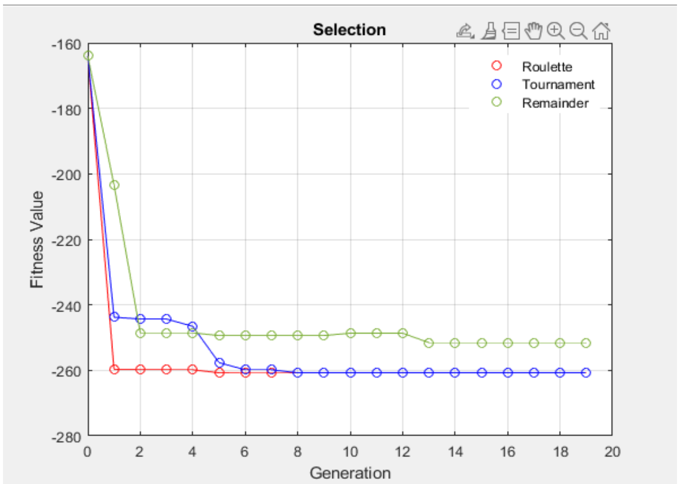

Figure 6. Selection options behaviour in seed 2.

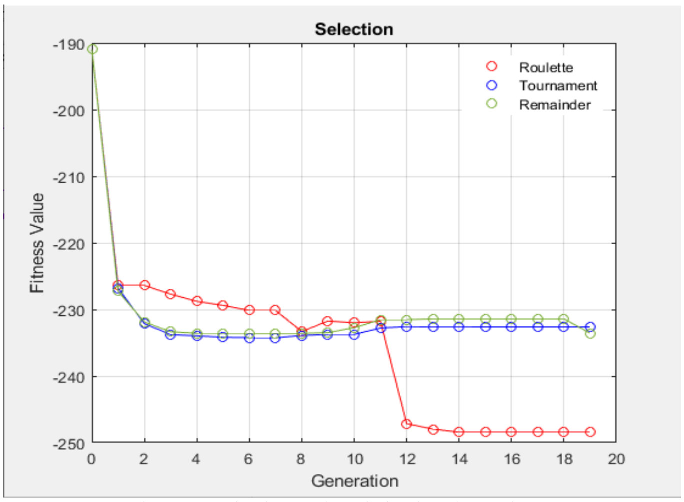

Figure 7. Selection options behaviour in seed 3.

The average fitness function values are (249.6), (250.1) and (245.3) for roulette, tournament and remainder selection respectively, these values are very close as average since all of selection options depends on a random or partially random selection. In each instance, each option behaves differently, which means that it cannot be said that an option is the best for each problem instance.

B. Crossover Operation

Crossover operation is the operation of combining the genetic information between two chromosomes, which yields in new chromosomes. It has different options such as scattered, heuristic, single point, and double point crossover. Scattered crossover creates random binary chromosomes and selects the genes where the chromosome is 1 from the first parent, and the genes where the chromosome is 0 from the second parent. Heuristic crossover increases the part taken from the parent with higher fitness value. Single and double points randomly create a new chromosome by crossing-over the parents in on or two points. Figures (8)-(10) show the behaviour of the different crossover options in the three different seeds.

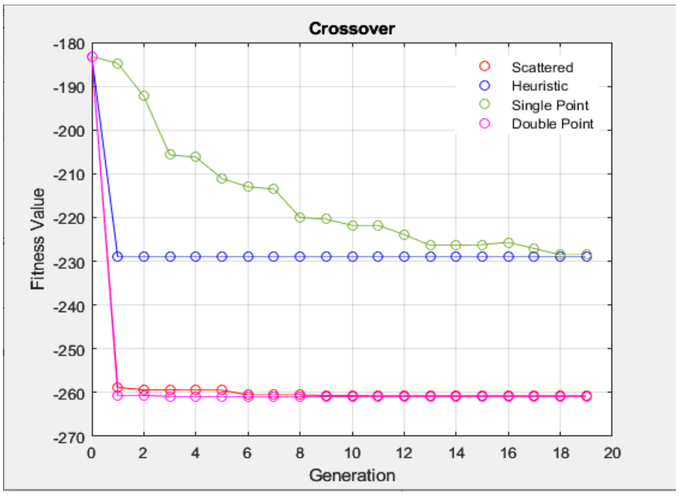

Figure 8. Crossover options behaviour in seed 1

The average fitness function values are (253.3), (220.6), (247.2), and (261) for scattered, heuristic, single point, and double point crossover respectively. Double point crossover shows the best fitness function values because it offers more exploration for the search space of the problem.

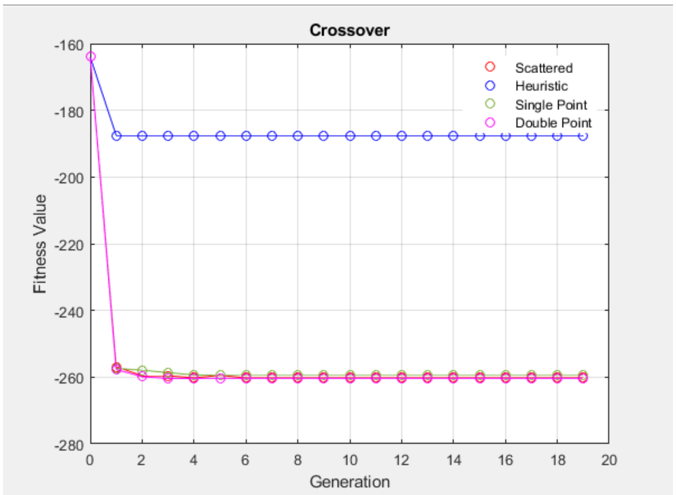

Figure 9. Crossover options behaviour in seed 2

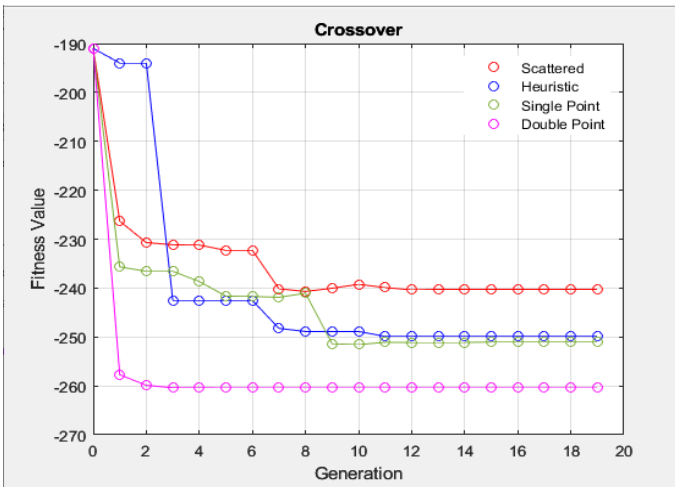

Figure 10. Crossover options behaviour in seed 3.

C. *Mutation Operation*

Mutation Operation is the process where a gene in a chromosome is changed to give a new solution. It has different options such as Gaussian, uniform, and power mutation. Gaussian mutation adds a random number taken from Gaussian distribution with mean 0 to each gene. Uniform mutation replaces a selected gene with values from selected range uniformly. Power mutation uses a specific equation to mutate.

Figures (11)-(13) show the behaviour of the different mutation options in the three different seeds.

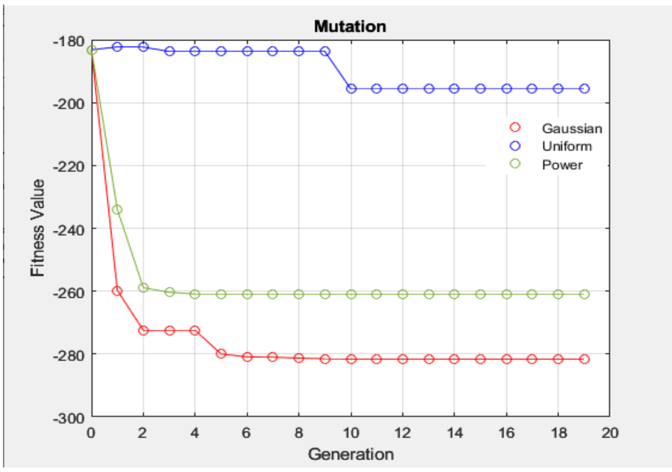

Figure 11. Mutation options behaviour in seed 1

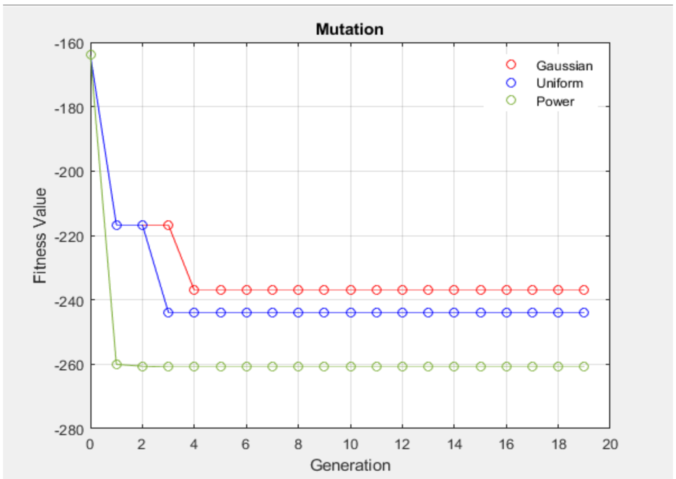

Figure 12. Mutation options behaviour in seed 2

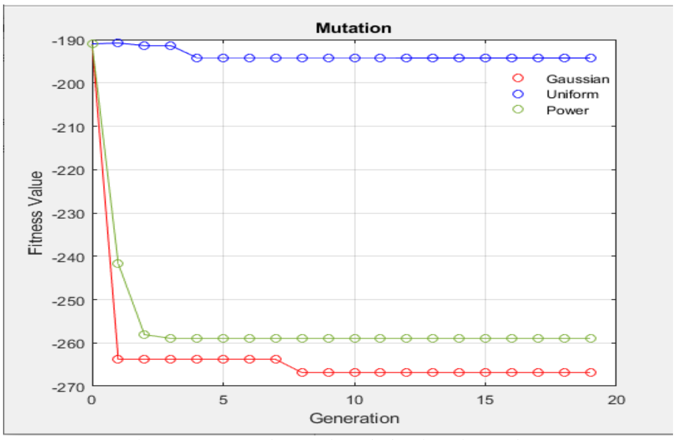

Figure 13. Mutation options behaviour in seed 3.

The average fitness function values are (259.7), (210.6), and (259.3) for Gaussian, uniform and power mutation respectively. Both Gaussian and power mutation resulted in good fitness function values. Gaussian mutation produces small, normally distributed changes, while power mutation introduces larger, uniform changes, as seen in figures (11)-(13), Gaussian mutation gradually improves the fitness value, and power mutation converged fast.

D. *Number of Elite Individuals*

Elites are the individuals in the current generation with the best fitness values. These individuals automatically survive to the next generation. Figures (14)-(16) show the behaviour of the different elite member's number in the three different seeds.

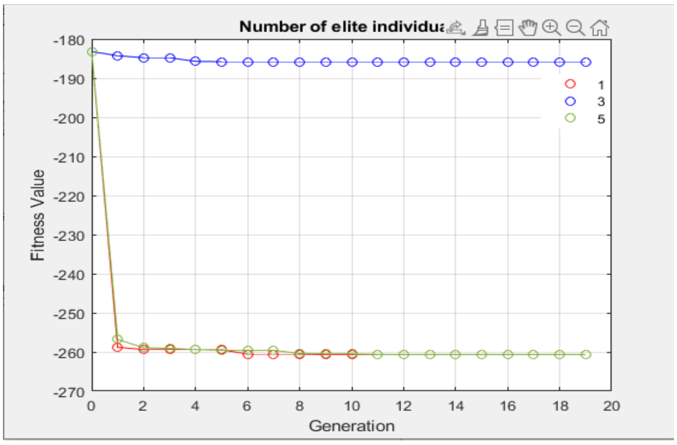

Figure 14. Changing the number of elites' behaviour in seed 1

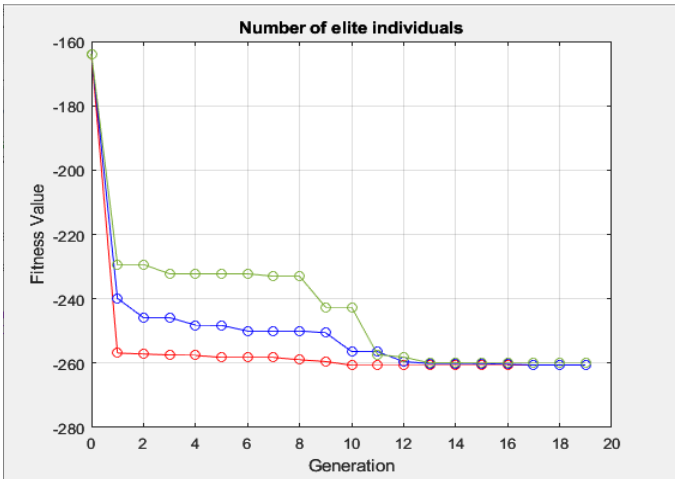

Figure 15. Changing the number of elites' behaviour in seed 2

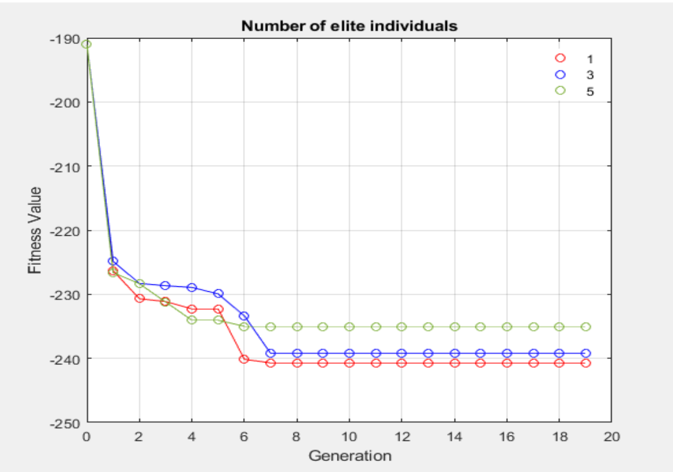

Figure 16. Changing the number of elites' behaviour in seed 3.

The average fitness function values are (253.3), (226.3), and (253.3) for 1, 3, and 5 elite individuals respectively. The more elite members means more exploitation. It cannot be said that a specific number of elite members is the best, because it depends on the initial population of the optimization problem.

E. Number of Individuals

The number of the individuals affect the population size, increasing the number of individuals expresses more exploration in the search space. Figures (17)-(19) show the behaviour of the different member's number in the three different seeds.

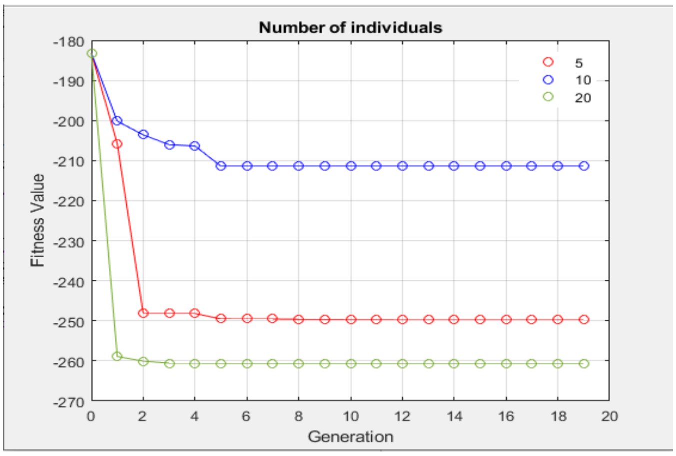

Figure 17. Changing the number of individual's behaviour in seed 1

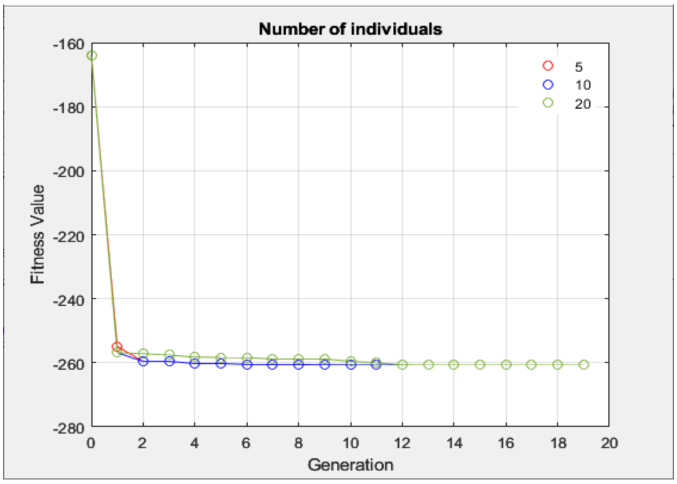

Figure 18. Changing the number of individual's behaviour in seed 2

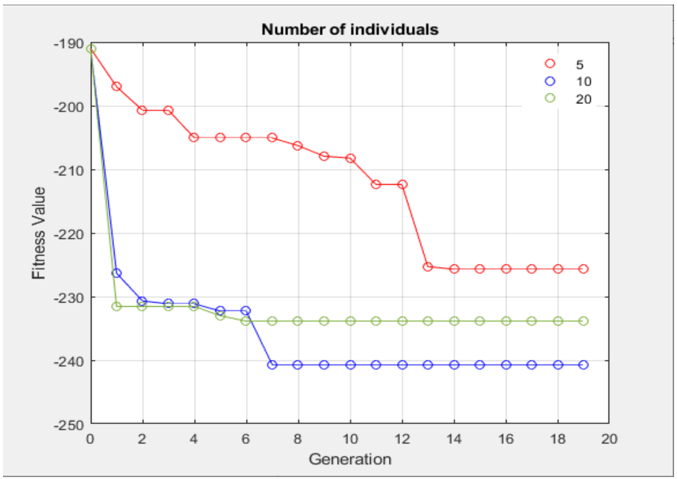

Figure 19. Changing the number of individual's behaviour in seed 3.

The average fitness function values are (244.67), (237), and (251.2) for 5, 10, and 20 individuals. For more number of individuals, the exploration will increase.

V. COMPARISON, EVALUATION & MATHEMATICAL PROOF

A. Comparison & Evaluation

The results from the past works are compared with the average values obtained from all simulations of the three seeds. The average value of the fitness functions is -234.87, and the average number of generations to converge is 3.63 vs -216.683 and 2 for the past work respectively. The evaluation formula is shown in equation (7).

$$\frac{ref\ [4]\ results - this\ paper\ results}{ref\ [4]\ results} \cdot 100\% \qquad (7)$$

Resulting in 5.41% better fitness value and 81.5% slower divergence for this paper results.

B. Mathematical Proof

Table (1) Shows the final values of the optimization variables and the value of the fitness function. The initial values chosen were $D = 0.76, V_{in} = 120\text{V}, V_{out} = 500\text{V}, \Delta i = 12, \Delta v = 14, R = 6\Omega$.

Parameter	Cf	Cc	Lf	Lc	f
Value	$3.08x10^{-8}$ F	$5.44x10^{-5}$ F	0.01 H	$9.1x10^{-5}$ H	41813 Hz

Table 1. Final values of the Algorithm.

Applying these values to equations (4)&(5):

$$Lc = \frac{0.76x120}{2x41813x12} = 9.1x10^{-5}\ H$$

$$Cc = \frac{0.76x500}{2x41813x6x14} = 5.41x10^{-5}\ F$$

Which are the same results in the output of the algorithm.

VI. Conclusion

This paper work resulted in better average fitness values (-234.87) vs (-216.683) and slower average convergence (3.63) vs (2), which is 81.5% slower compared with past works. The best instance was using the Gaussian mutation in seed 1, which resulted in (-280) fitness value and converged in 5 generations.

The genetic algorithm outcomes are very sensitive to the initial population, exploration and exploitation rates used in each run, it was shown that the same options for different initial populations resulted in an unexpected behaviour compared to the other options. This concludes that it cannot be said that a genetic operation or option is superior to other genetic operation for all instances, it is only true for a set of instances.

References

[1] Mamizadeh, Ali, Naci Genc, and Ramin Rajabioun. "Optimal tuning of pi controller for boost dc-dc converters based on cuckoo optimization algorithm." 2018 7th international conference on renewable energy research and applications (ICRERA). IEEE, 2018.

[2] Çelebi, Mehmed. "Efficiency optimization of a conventional boost DC/DC converter." Electrical Engineering 100 (2018): 803-809.

[3] Sörensen, Kenneth, and Fred Glover. "Metaheuristics." Encyclopedia of operations research and management science 62 (2013): 960-970.

[4] Kumaraguru Prabakar, F. L. (2012). Application of genetic algorithm for the improved performance of boost converters. *IFAC Proceedings Volumes*, 85-90.

[5] Mendez, E. (2020). Novel design methodology for DC-DC converters applying metaheuristic optimization for inductance selection. *Applied Sciences 10.12*, 4377.

[6] Giuseppe Marsala, A. R. (2018). Reliability and efficiency optimization assisted by genetic algorithm to design a quadratic boost DC/DC converter. *IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*.

[7] Ali Mamizadeh, N. G. (2018). Optimal tuning of pi controller for boost dc-dc converters based on cuckoo optimization algorithm. *IEEE*.

[8] Mirjalili, Seyedali, et al. "Genetic algorithm: Theory, literature review, and application in image reconstruction." Nature-Inspired Optimizers: Theories, Literature Reviews and Applications (2020): 69-85.

[9] Erickson, Robert W., and Dragan Maksimovic. Fundamentals of power electronics. Springer Science & Business Media, 2007.

[10] MATLAB. (2022). *Genetic Algorithm*. Retrieved from MathWorks: https://www.mathworks.com/help/gads/how-the-genetic-algorithm-works.html