ELECTRICAL ENGINEERING AND ENERGY

OPEN ACCESS

A Comparative Analysis of Genetic Algorithm Replacement Strategies in a Real-World Optimization Problem: An Elitism and Diversity Oriented Perspective

Hüseyin Doğan

Department of Mechatronics Engineering, Faculty of Technology, Selcuk University, Konya, Türkiye

Submitted: 10 April 2025
Accepted: 29 April 2025
Published: 30 April 2025

Corresponding author
Hüseyin Doğan,
huseyindogan@selcuk.edu.tr

Abstract: In this study, generation replacement strategies used in genetic algorithms are comparatively analysed in the context of a real-world optimization problem, with a focus on the balance between elitism and diversity. The problem under consideration involves solving the equations of the Selective Harmonic Elimination (SHE) method, which is widely used for controlling Multilevel Inverters (MLIs). Six algorithms, each employing a different replacement mechanism, were tested under four distinct scenarios composed of various population sizes and iteration counts. The results were evaluated using fundamental statistical metrics, boxplots, and convergence curves. The findings reveal that elitist strategies perform better in large-population, long-duration scenarios, whereas approaches prioritizing diversity yield more effective results under limited resource conditions. This study systematically demonstrates the impact of different replacement strategies on optimization performance and offers valuable insights for strategy selection in real-world optimization problems.

Keywords Genetic Algorithm, Replacement Strategy, Elitism, Diversity, Selective Harmonic Elimination (SHE), Multilevel Inverter (MLI)

1. Introduction

Genetic Algorithm (GA) is a population-based metaheuristic optimization method inspired by evolutionary processes observed in nature. Originally introduced by John Holland (Holland, 1992), this approach aims to solve a given optimization problem through an iterative computational process based on biological mechanisms such as natural selection, genetic crossover, and mutation. One of the most critical aspects of the algorithm is the establishment of a fitness evaluation mechanism that mimics the evolutionary principle wherein individuals better adapted to environmental conditions have a higher chance of survival and gene propagation. Accordingly, a fitness (objective) function must be defined in a way that both satisfies system requirements and respects problem-specific constraints. Individuals in the population are evaluated throughout various stages of the algorithm based on the values obtained from this function.

The algorithm begins with a population composed of individuals, each representing a potential solution to the given problem, and the initial fitness scores of these individuals are calculated based on the defined objective function. Following this initial stage, the iterative process begins. At the start of each iteration, a subset of individuals is selected based on their fitness scores. These selected individuals are then subjected to small variations through various genetic operators in order to explore potentially better solutions.

By the end of the iteration, a new generation is formed, which is expected to contain better alternatives in terms of overall solution quality. This new generation is referred to as the "offspring" while the generation at the beginning of the iteration is called the "parents". The fitness scores of the offspring are recalculated, and the individuals are reassigned as parents for the next iteration. This process continues until a satisfactory solution is reached or a predefined number of iterations is completed.

GAs have been widely applied across various fields—from engineering to economics—due to their strong global search capability and effectiveness in solving complex problems that are multimodal, nonlinear, or subject to multiple constraints. This broad range of applications is evident in numerous studies, including optimal power flow in electrical grids (Bakirtzis et al., 2002; Osman et al., 2004), parameter tuning in control systems (J. Zhang et al., 2009), feature selection in machine learning (Halim et al., 2021; Smith & Bull, 2005), scheduling and routing optimization in manufacturing systems (Gen & Cheng, 1999), financial forecasting (Kim & Han, 2000) and supply chain and inventory management (Disney et al., 2000).

The generation update phase performed at the end of each iteration ("Survivor Selection" or "Replacement") is not explicitly defined in the structure of the classical GA. More precisely, in the standard approach, no special method is applied during this phase; instead, the new generation (offspring) simply replaces the previous one (parents). However, this can result in the loss of potentially high-quality individuals from the earlier population, thereby slowing down the overall optimization process. To address this issue, various methods have been proposed in the literature in which certain individuals from the previous generation are preserved and transferred to the next generation either directly or through a specific selection process. These approaches are generally classified under the concept of "elitism" (Du et al., 2018).

Studies in the literature generally have focused on the key components of genetic algorithms—such as selection, crossover, and mutation operations—that significantly affect their performance (Alhijawi & Awajan, 2024; Ali et al., 2020; Hassanat et al., 2019; Hong et al., 2002; Katoch et al., 2021; F. Zhang et al., 2008). In contrast, generation update strategies are often treated as a fixed structural component and are rarely subjected to detailed analysis. In several existing studies, replacement methods are typically introduced as part of a newly proposed GA variant (Ahn & Ramakrishna, 2003), rather than being independently evaluated or compared with alternative approaches. In the limited number of studies that do involve comparisons, evaluations are mostly conducted using standard test functions (Brouwer et al., 2022), which often fail to reflect the dynamic and constrained nature of real-world optimization problems.

In this study, the effects of different replacement strategies used in genetic algorithms were systematically compared in the context of solving the equations of the Selective Harmonic Elimination (SHE) method, which represents a real-world optimization problem. SHE is one of the most widely used control techniques in multilevel inverters (MLIs). The primary goal of this method is to eliminate specific harmonic components in the inverter output voltage while keeping the fundamental harmonic amplitude as close as possible to a desired reference value. To achieve this, mathematical expressions are derived for the relevant harmonics, where the switching angles are defined as variables. Satisfying all objectives simultaneously requires determining the optimal set of switching angles that solve these equations concurrently (Yang et al., 2017). However, due to the nonlinear structure and trigonometric nature of the equations, solving SHE becomes increasingly challenging as the number of inverter levels rises.

Within this scope, six genetic algorithm variants were investigated; one with a standard structure and no replacement applied, and the remaining five employing different generation update mechanisms. The algorithms were tested under four different scenarios composed of various combinations of population sizes and iteration counts, with independent runs conducted for each. The results were evaluated using key statistical metrics such as Best, Median, Worst, and Std (standard deviation), along with boxplot visualizations. Additionally, the solutions obtained were applied to the switching devices of a multilevel inverter modelled in MATLAB/Simulink, and the algorithms were further assessed based on fundamental component error and harmonic suppression performance in the output voltage.

The findings of this study reveal that while elitist strategies used during the replacement phase positively impact GA performance, they also pose a risk of reducing population diversity in scenarios with limited resources, such as low population sizes and restricted iteration counts.

In such cases, the rapid dominance of elite individuals within the population may lead to the premature elimination of individuals with relatively poor fitness but potential for long-term convergence to the global optimum. As a result, the algorithm's overall exploratory ability weakens, and the likelihood of getting trapped in local minima increases. On the other hand, diversity-oriented approaches help mitigate this risk under resource-constrained conditions and enable a more effective exploration of the solution space. These results indicate that the replacement strategy chosen should not only preserve high-quality individuals but also maintain a balanced structure that sustains diversity depending on the problem environment.

2. Switching Angle Determination via SHE in CHB-MLI Systems

In conventional two-level inverters, the output voltage is generated by switching between only two fixed levels, typically +Vdc and –Vdc. In such systems, power switches are operated at high switching frequencies to rapidly alternate the output voltage between these two levels, aiming to approximate an ideal sinusoidal waveform. While this configuration is simple and cost-effective—due to its ability to operate with a single DC source and requiring fewer semiconductor components—it also presents some significant disadvantages (Bertin et al., 2023). Since the switches must handle high voltage levels during every transition, the lifespan of components is reduced, and the overall reliability of the system is compromised. Moreover, high switching frequencies considerably increase switching losses and electromagnetic interference (EMI).

To overcome the issues observed in conventional two-level inverter structures, various MLI topologies have been developed (El-Hosainy et al., 2017). The core approach of these topologies is to generate the output voltage in a stepwise manner using multiple intermediate voltage levels, thereby producing a waveform that resembles a staircase. As a result, semiconductor devices switch lower voltage levels in each cycle, and this switching occurs only twice per half-cycle. Consequently, the need for high switching frequencies is reduced, which significantly minimizes losses, voltage stress on components, and EMI

Among MLI topologies, one of the most widely adopted structures is the Cascaded H-Bridge Multilevel Inverter (CHB-MLI) (Sunddararaj et al., 2020). In this topology, multiple H-bridge inverter modules—each with identical specifications and supplied by an independent DC source—are connected in series. Each module contributes one voltage level to the staircase-shaped output waveform (Dahidah et al., 2015). As the number of modules increases, the number of steps in the waveform also increases, resulting in an output voltage that more closely approximates a sinusoidal form. In addition to the conventional advantages offered by multilevel inverters, the CHB-MLI topology provides a significant benefit in terms of structural modularity. Thanks to this modularity, the system can be easily reconfigured by adding or removing a module when a change in output voltage level is required. Furthermore, in the event of a fault or maintenance need, the affected module can be isolated and replaced without interfering with the entire system. This reduces maintenance time, simplifies servicing, and enhances overall system reliability.

In the CHB-MLI topology, the output voltage waveform is defined by the switching angles that determine when each module contributes to the output. Since each module generates one voltage level in the staircase waveform, the number of switching angles corresponds to the number of modules used. Therefore, a system with n modules requires n switching angles $(\theta_1, \theta_2, \ldots, \theta_n)$. To accurately compute these angles, n independent equations are needed. method provides a systematic approach for determining these angles (Lingom et al., 2024). Among the resulting n equations, one is formulated to match the fundamental component of the output voltage to the desired reference value, while the remaining $n - 1$ equations are each constructed to eliminate a specific harmonic component. As a result, harmonic control in the SHE method emerges both as a direct objective and as a natural outcome of the switching angle calculation process

In this study, the CHB-MLI configuration under consideration is a single-phase, 11-level inverter. Accordingly, five ($n = 5$) H-bridge inverter modules are employed, each supplied by a DC voltage source of 50 V. The output voltage waveform presented in Figure 1 illustrates the stepwise structure formed by the contribution of each module. Here, V_x ($x = 1,2,3,4,5$) represents the output voltage of the corresponding module. Each module is activated at its respective switching angle θ_x adding its voltage to the existing total and thereby increasing the output voltage to the next level. This process continues sequentially during each quarter-cycle, resulting in the desired multilevel voltage profile.

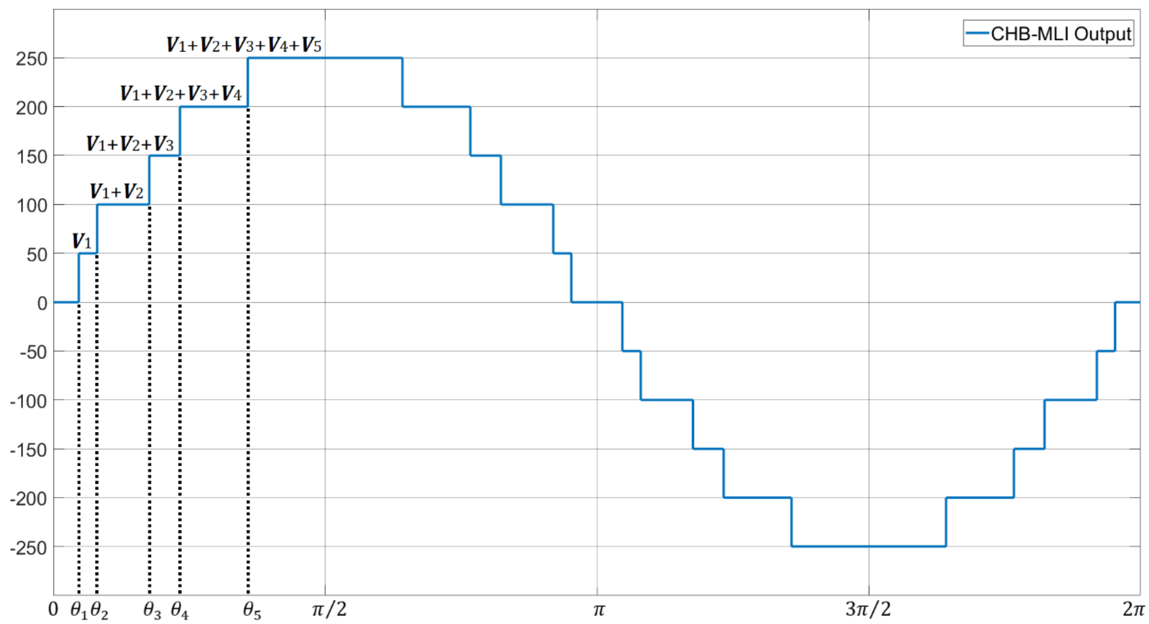


Figure 1 The Output Voltage Waveform for a Single-Phase CHB-MLI

Under the assumption of half-wave symmetry, even-order harmonic components are inherently eliminated from the output voltage waveform as shown in the figure. Furthermore, assuming that all DC sources are equal, a general expression can be defined for each odd harmonic component present in the output voltage. In this equation, h and V_h represent the order of the odd harmonics in the output voltage ($h = 1,3,5 \ldots$) and the amplitude of the h^{th} harmonic component, respectively. θ_i denotes the i^{th} switching angle ($i = 1,2,3,4,5$).

$$V_h = \frac{4V_{DC}}{h\pi} \sum_{i=1}^{5} \cos(h\theta_i) \quad (1)$$

Within the scope of the SHE method, an algebraic system of equations is derived from the general expression above to regulate the desired fundamental component level in the output voltage while suppressing undesired harmonics. These equations represent the amplitudes of the harmonic components expressed as functions of the switching angles. Among them, one corresponds to the fundamental component ($h = 1$), while the remaining four represent the amplitudes of the targeted harmonics to be eliminated ($h = 3,5,7,9$).

The primary objective of the optimization algorithms examined in this study is to numerically solve this system of equations to determine valid switching angles that satisfy the defined criteria.

3. Generation Update Strategies in Genetic Algorithms

In genetic algorithms, generation replacement (also referred to as survivor selection) is a critical phase that determines how the newly generated population at the end of an iteration is transferred to the next generation. This process plays a direct and essential role in preserving genetic diversity, preventing premature convergence, and improving overall solution quality. Replacement strategies are based on different rules regarding how the parent and offspring populations are compared, which individuals are preserved, and which are eliminated.

In this study, several alternative replacement strategies to the classical approach were evaluated, and their effects on the SHE problem were comparatively analyzed. Within this scope, a standard real-coded GA framework was employed across all variants to ensure consistency. Each individual X_i in the population of size N was defined as a solution vector composed of five switching angles θ_j $(j = 1,2,\dots,5)$, as expressed in Equation 2. These angles must satisfy the constraint provided in Equation 3. In all algorithms, the selection process was carried out using the roulette wheel method, single-point crossover was applied during the crossover phase, and a basic mutation strategy was used.

$$X_i = (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5)\ \ i \in \{1,2,\dots,N\} \tag{2}$$

$$0 \leq \theta_1 < \theta_2 < \theta_3 < \theta_4 < \theta_5 \leq \pi/2 \tag{3}$$

In the algorithms under investigation, the fitness function derived from the SHE equations was defined in Equation 4 to evaluate the individuals in the population (Bektaş et al., 2023). This function is formulated based on a two-component error structure. The first term, $f_1(X_i)$, represents the difference between the obtained fundamental component amplitude and the desired reference value (V_{1ref}). The second term $f_2(X_i)$, evaluates the suppression level of undesired harmonic components corresponding to the 3rd, 5th, 7th and 9th harmonic orders. During this evaluation, weighting coefficients ($w = 2.5, 2, 1.5, 1$) were applied to assign greater importance to lower-order harmonics, making them a higher priority in the optimization process. As a result, minimizing the fitness value leads to a solution where the fundamental component closely matches the target value while the unwanted harmonics are effectively suppressed.

$$f(X_i) = \left(f_1(X_i) + f_2(X_i)\right) \tag{4}$$

$$f_1(X_i) = \left| V_{1ref} - \frac{4V_{DC}}{\pi} \sum_{j=1}^{5} \cos\left(X_{i,j}\right) \right| \tag{5}$$

$$f_2(X_i) = \left(\frac{4V_{DC}}{\pi}\right)^2 \sum_{k=1}^{4} w_k \left(\frac{1}{h_k} \sum_{j=1}^{5} \cos\left(h_k X_{i,j}\right) \right)^2 \tag{6}$$

$$h = 3, 5, 7, 9\ \ w = 2.5, 2, 1.5, 1$$

The first algorithm included in the comparison, GA01, is the standard genetic algorithm without any replacement strategy applied. In this structure, the offspring individuals ($X_{off}(t)$) generated through selection, crossover, and mutation in each iteration are directly assigned as the parents of the next iteration ($X_{par}(t+1)$), as shown in Equation 7, while the current iteration's parents ($X_{par}(t)$) are discarded. As a result, the population is completely replaced in every generation. This technique is commonly referred to in the evolutionary algorithms literature as the (μ, λ) selection strategy.

$$X^{par}(t+1) = X^{off}(t), F^{par}(t+1) = F^{off}(t) \tag{7}$$

The second algorithm compared (GA02) corresponds to the structure commonly known in the literature as the elitist GA. In this method, at the beginning of each iteration, the n best-performing individuals based on fitness are selected and stored as elite individuals X^e (Equation 8). At the end of the iteration, the n worst-performing individuals X^w among the offspring are identified (Equation 9) and directly replaced with the previously saved elites (Equation 10). After this replacement, the updated offspring population—including the elites—is used as the parent population for the next iteration, as in the standard GA. In this approach, the elite individuals are transferred to the next generation without being subjected to any additional comparison or selection mechanism.

$$\begin{gathered} X^e(t) = \{X_1^e, X_2^e, \dots, X_n^e\} \\ f_1^e \leq f_2^e \dots \leq f_n^e \end{gathered} \tag{8}$$

$$\begin{gathered} X^w(t) = \{X_1^w, X_2^w, \dots, X_n^w\} \\ f_1^w \geq f_2^w \dots \geq f_n^w \end{gathered} \tag{9}$$

$$X_k^w \leftarrow X_k^e \quad k = 1,2, \dots, n \tag{10}$$

The third algorithm (GA03) is based on a tournament-based replacement strategy with random pairing. In this method, at the end of each iteration, a tournament is conducted between the parent population and the newly generated offspring population. For this purpose, as defined in Equation 10, an index array is created by randomly selecting non-repeating integers from the range $[1, N]$ to represent the offspring individuals. Then, each parent individual $X_i^{par}(t)$ is paired and compared with the corresponding offspring individual $X_k^{off}(t)$ from the index array, as shown in Equation 11. The individual with the better fitness value is selected for transfer to the next generation.

$$p = randperm(N) \tag{11}$$

$$X_i^{par}(t+1) = \begin{cases} X_i^{par}(t), & f_i^{par}(t) < f_k^{off}(t) \\ X_k^{off}(t), & f_k^{off}(t) > f_i^{par}(t) \end{cases} \quad i = 1,2, \dots, N \ (k = p_i) \tag{12}$$

The fourth algorithm (GA04) adopts a roulette wheel-based replacement strategy. In this structure, the current parent population $X^{par}(t)$ and the newly generated offspring population $X^{off}(t)$ are combined into a unified pool $X^{com}(t)$ consisting of $2N$ individuals (Equation 13). For each individual in this pool, selection probabilities p_i are calculated by taking the inverse of their fitness values and normalizing them, followed by computation of cumulative selection probabilities pk_i (Equation 14). Using these cumulative probabilities, individuals corresponding to randomly generated numbers in the range $[0,1]$ are selected, and a new parent population $X_i^{par}(t+1)$ consisting of N individuals is formed (Equation 15)

$$X^{com}(t) = \left(X^{par}(t) \cup X^{off}(t)\right) \tag{13}$$

$$\begin{gathered} p_i = \frac{1/f_i^{com}}{p_{sum}} \quad p_{sum} = \sum_{i=1}^{2N} 1/f_i^{com} \\ pk_i = \sum_{x=1}^{i} p_x \end{gathered} \quad i = 1,2, \dots, 2N \tag{14}$$

$$X_i^{par}(t+1) = \begin{cases} X_1^{com} & 0 < rand \leq pk_1 \\ X_2^{com} & pk_1 < rand \leq pk_2 \\ \vdots & \vdots \\ X_{2N}^{com} & pk_{2N-1} < rand \leq 1 \end{cases} \quad i = 1,2,\dots,N \tag{15}$$

The fifth algorithm (GA05) is based on a ranking-based elitist replacement strategy. In this method, as shown in Equation 13, the parent population and the newly generated offspring population are merged into a common pool. The individuals in this pool are then ranked according to their fitness values (Equation 16), and the best N individuals are directly selected to form the next generation population $X_i^{par}(t+1)$ (Equation 17). This strategy does not involve any comparison, probability, or selection mechanism; instead, it relies solely on absolute fitness ranking. This approach corresponds to the $(\mu + \lambda)$ selection strategy commonly used in evolutionary algorithms.

$$f_1^{com} \leq f_2^{com} \dots \leq f_{2N}^{com} \tag{16}$$

$$X_i^{par}(t+1) = X_i^{com}(t) \quad f_i^{par}(t+1) = f_i^{com}(t) \qquad i = 1,2,\dots,N \tag{17}$$

The sixth and final algorithm (GA06) implements a partial elitism-based replacement strategy that aims to preserve the best-performing individuals from both the parent and offspring populations while also giving a chance to lower-performing individuals from both generations. In this method, the current parent and offspring populations are first ranked separately according to their fitness values (Equation 18). From each population, q individuals ($q = N/4$) in the top quartile are selected to form the subsets $X^1(t)$ and $X^2(t)$, respectively (Equation 19). Additionally, q individuals are randomly selected from the remaining lower-performing individuals in each population (from the range $[q+1, N]$), forming subsets $X^3(t)$ and $X^4(t)$ (Equation 20). Finally, the union of these four subsets constitutes a combined population $X^{com}(t)$ of N individuals, which is assigned as the parent population $X^{par}(t+1)$ for the next generation (Equation 21). This approach can be considered a hybrid structure that simultaneously promotes elitism and diversity and may be viewed as a variation of the previous replacement strategy.

$$f_1^{par} \leq f_2^{par} \dots \leq f_N^{par}, f_1^{off} \leq f_2^{off} \dots \leq f_N^{off} \tag{18}$$

$$X_i^1(t) = X_i^{par}(t), X_i^2(t) = X_i^{off}(t) \qquad i = 1,2,\dots,q \quad (q = N/4) \tag{19}$$

$$\begin{aligned} X_i^3(t) &= \{X_{r1}^{par}, X_{r2}^{par}, \dots, X_{rq}^{par}\} && rx \in \{q+1, q+2, \dots, N\} \\ X_i^4(t) &= \{X_{r1}^{off}, X_{r2}^{off}, \dots, X_{rq}^{off}\} && x = 1,2,\dots,q \end{aligned} \tag{20}$$

$$X^{com}(t) = \left(X^1(t) \cup X^2(t) \cup X^3(t) \cup X^4(t)\right) \rightarrow X^{par}(t+1) = X^{com}(t) \tag{21}$$

Generation replacement strategies are designed to prevent the loss of high-quality solutions. While the preservation of good individuals is commonly referred to as elitism in the literature, this approach also introduces the risk of gradually losing population diversity; one of the fundamental driving forces of evolutionary algorithms. Therefore, the balance between elitism and diversity plays a critical role in optimization problems. In this context, the algorithms examined in this study are evaluated with respect to their approaches to elitism and diversity as follows:

- GA01, which does not incorporate any elitism mechanism, completely eliminates parent individuals after each iteration, regardless of their quality. Diversity is maintained solely through the effect of the variation operators applied to the offspring population.
- The only difference between GA02 and the previous algorithm is the preservation of a small elite portion (5%) of the parent population. This limited intervention introduces a mild level of elitism

into the optimization process. However, since elite individuals are carried over to the next generation without any comparison or competition, this may result in the elimination of potentially superior offspring in the later stages of long-term iterations, where differences among individuals diminish. Apart from this, diversity is maintained (as in GA01) only through the offspring, with no contribution from the previous generation.

- The tournament structure in GA03 ensures that individuals with better (i.e., lower) fitness values are largely preserved in the population, indicating a strong elitist characteristic. However, because pairings are made randomly, there is always a possibility that two weak individuals may compete against each other. This helps prevent diversity from being completely lost and allows the partial retention of individuals who, despite low current performance, may hold potential in the evolutionary process.

- GA04, which uses a roulette wheel-based selection, combines the parent and offspring populations into a shared pool, where individuals with better fitness values have a much higher probability of being selected for the next generation. Moreover, due to the stochastic nature of this approach, top-performing individuals may be selected multiple times, increasing the risk of population dominance by a few elites. Although the probability of selecting low-performing individuals is not zero, it remains very low. Thus, compared to the tournament method, GA04 supports diversity to a lesser extent while applying elitism in a stricter and less controlled manner.

- The principle of GA05 is purely deterministic, with no probabilistic components involved. After merging the parent and offspring populations, the top N individuals are directly selected to form the next generation. The selection of the best individuals is guaranteed with 100% certainty, making GA05 more stable and reliable in terms of elitism compared to probabilistic approaches like roulette or tournament selection. However, because the selection is based on absolute ranking, even the best individual is chosen only once. This ensures that elitism is implemented effectively yet in a balanced way. As a result, while elite individuals are preserved, moderately fit individuals also have a chance to remain in the system helping to sustain diversity. That said, this strategy is intolerant toward weaker individuals: those with the worst fitness values are categorically excluded from the next generation.

- GA06 aims to consciously balance elitism and diversity. As in GA05, the transfer of elite individuals is guaranteed, making the elitist aspect of this approach deterministic. However, unlike GA05, the elite individuals are selected separately from both the parent and offspring populations, and their total proportion is lower. Moreover, diversity is not left to the natural course of the evolutionary process; instead, it is actively promoted by applying positive selection bias in favour of weak individuals. While this guarantees the preservation of diversity, it also deliberately restricts the spread of elite solutions. Consequently, in long-term iterations, the suppressing effect on elite individuals may pose a risk of stagnation in solution quality.

4. Results and Discussions

In this study, the performance of six different GA variants employing distinct replacement strategies was compared in the context of solving the SHE equations. For this purpose, each algorithm was tested under four different scenarios. These scenarios were structured based on combinations of population size (N) and number of iterations (T), as outlined below:

(i) $N = 20,\ T = 20$; (ii) $N = 20,\ T = 100$; (iii) $N = 100,\ T = 20$; (iv) $N = 100,\ T = 100$.

The primary objective of designing this scenario structure is to evaluate how the algorithms perform under both resource-constrained conditions (low N and T) and conditions that allow for broader search capabilities (high N and T).

For each algorithm, 501 independent runs were performed in each scenario. In these runs, the objective function defined in Equation 4 was used to evaluate the quality of the solutions. The reference fundamental component amplitude (V_{1ref}) was set to 250 V. In each run, the best (i.e., lowest) fitness value achieved by the algorithm was recorded. From the resulting set of 501 values, the performance metrics Best, Median, Worst, and Std (standard deviation) were computed and comparatively presented in the relevant tables (Tables 1-4)). To compare the distributions of these values, boxplots (Figures 2, 4, 6, 8) were included, and convergence curves (Figures 3, 5, 7, 9) were used to analyse the average progress trends of the algorithms throughout the iterations.

The results for the first scenario are presented in Table 1 and Figures 2 and 3. This scenario represents a resource-constrained setting in which both the population size and the number of iterations are kept low. The limited number of iterations prevents the algorithms from fully demonstrating their optimization capabilities. Therefore, the ability to perform a fast and effective exploration of the search space becomes critically important. In this context, the small population size places the entire burden of exploration on the algorithm's capacity to maintain diversity.

Table 1 Optimization results for N=20 and T=20

Algorithms	Statistical Summary of Fitness Values			
	Best	**Median**	**Worst**	**Std**
GA01	12.2832	85.3245	269.2885	47.5573
GA02	6.9794	53.5927	248.0116	44.6122
GA03	5.8093	59.8713	250.0669	59.3952
GA04	6.5139	75.7216	266.1919	57.9014
GA05	5.5482	88.4709	268.7646	61.6239
GA06	7.6176	48.1857	178.0961	28.7910

According to the results, GA06, which enforces diversity structurally by design, stands out as the most successful algorithm in terms of both performance metrics and boxplot appearance. It is followed by GA02, which allows diversity to emerge naturally while applying a moderate and controlled level of elitism. Although the Median values of these two algorithms are close, GA06 exhibits a significantly lower Std value, indicating more consistent and stable outcomes.

GA03, which partially allows weaker individuals to survive through tournament selection, ranks just behind these two. GA04, while not as tolerant to diversity as GA03, still offers more flexibility than GA05, placing it next in line. GA05, where elitism is dominant and diversity is nearly disregarded, performs similarly to the standard GA01, and in some cases even worse. Notably, the convergence curves in Figure 3b clearly show that the final performance ranking of the algorithms aligns precisely with the extent to which each strategy supports diversity.

The results for the second scenario are presented in Table 2 and Figures 4 and 5. This scenario represents a longer evolutionary process combined with a small population size. This combination is particularly valuable for observing the algorithms' abilities to maintain diversity and their long-term convergence behaviour.

Table 2 Optimization results for N=20 and T=100

Algorithms	Statistical Summary of Fitness Values			
	Best	**Median**	**Worst**	**Std**
GA01	6.5454	35.2412	99.8045	19.7326
GA02	4.6600	15.2062	54.7889	13.4916
GA03	4.5360	14.8900	115.6199	18.8921
GA04	5.6908	30.5692	104.0617	18.7601
GA05	4.5427	10.1239	111.7354	21.6504
GA06	5.0021	16.4486	70.5006	13.8457

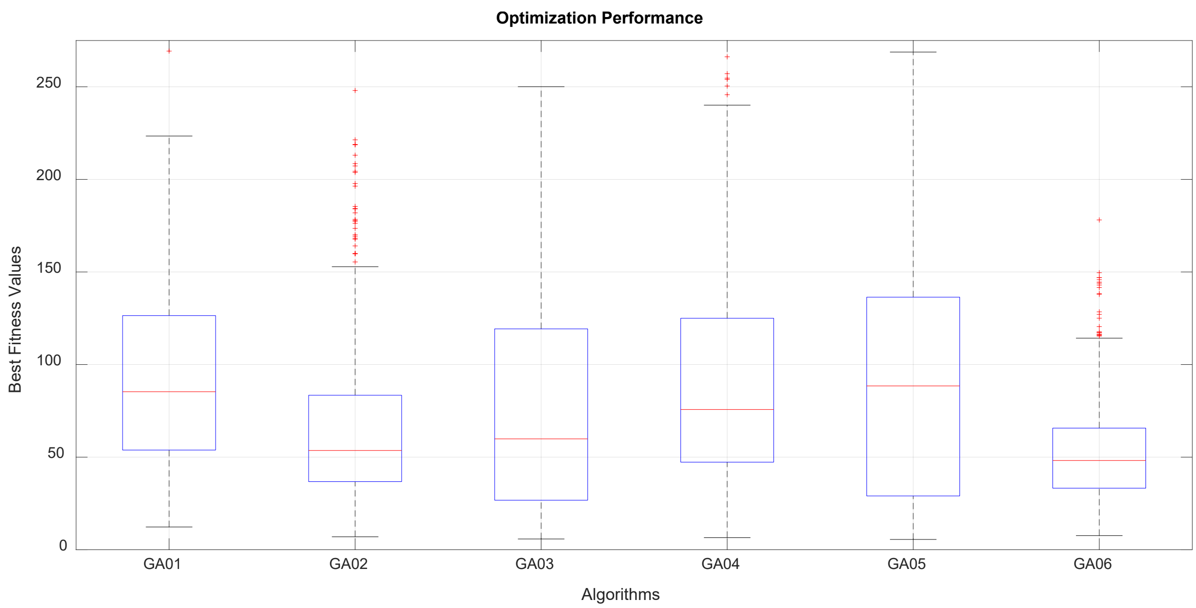


Figure 2 Boxplot graphic of the algorithms for N=20, T=20

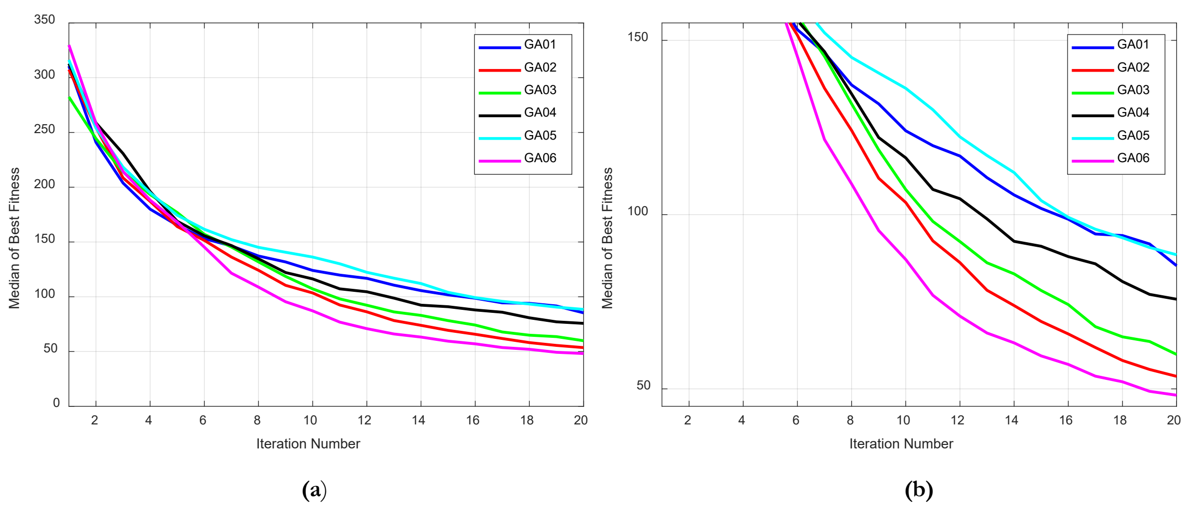


Figure 3 Convergence curves of the algorithms for N=20, T=20; **(a)** normal, **(b)** magnified view

The results indicate that, compared to the first scenario, all algorithms achieved lower fitness values and the performance differences between them became more pronounced. Analysing the boxplots and statistical metrics, it is evident that the increased number of iterations—despite a fixed population size—enabled GA05 to reveal its potential. GA05, which attained the lowest Median value by a significant margin, is followed by GA03, GA02, and GA06, respectively. However, the differences in Median values among these three algorithms are statistically negligible.

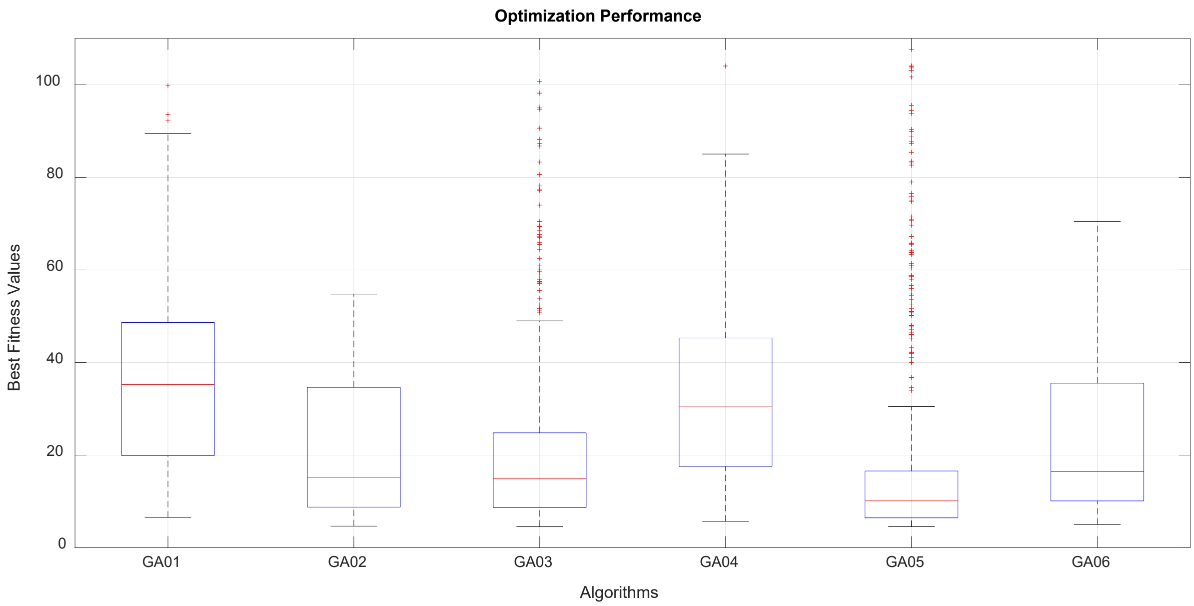
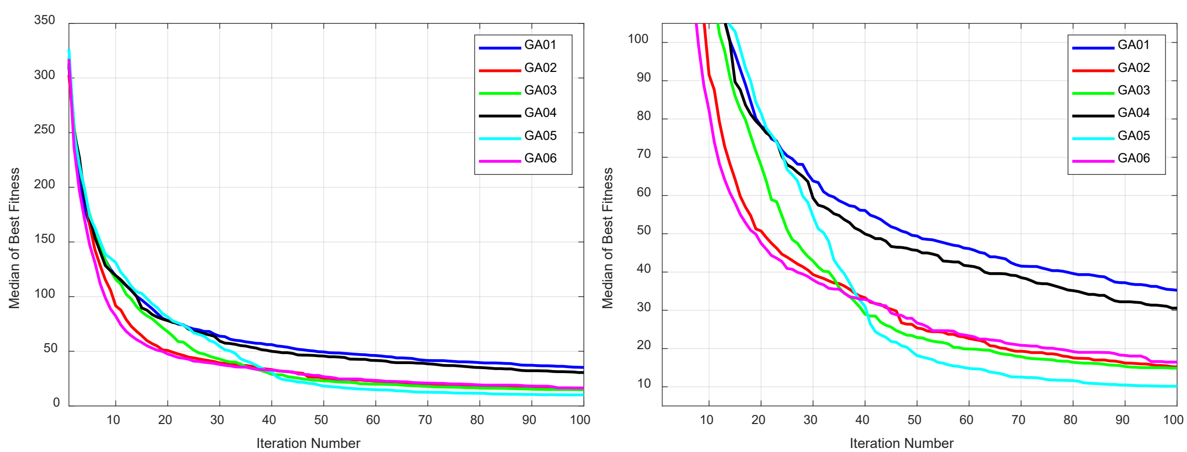


Figure 4 Boxplot graphic of the algorithms for N=20, T=100



(a) **(b)**

Figure 5 Convergence curves of the algorithms for N=20, T=100; **(a)** normal, **(b)** magnified view

The superiority previously achieved by GA06 and GA02 due to their diversity-oriented designs appears to have relatively diminished in this scenario, while algorithms with stronger elitist structures have begun to take the lead. Nonetheless, the balanced distribution in the boxplots and the low Std values of these two algorithms indicate that the positive influence of diversity is still present.

On the other hand, although GA03 and GA05 performed well overall, their results include a substantial number of outliers, which contributes to their relatively high standard deviation values. This suggests that under the current population size, elitist strategies have not yet achieved full effectiveness in eliminating low-quality individuals.

GA04, meanwhile, lagged behind these algorithms and exhibited a performance comparable to that of GA01.

The convergence curves clearly reveal the optimization superiority of GA05 and the improved performance of GA03 compared to the previous scenario. Although GA06 and GA02 initially take the lead thanks to the advantages of diversity, they are eventually overtaken—first by GA03, and then by GA05. In the final phase of the iteration process, GA05 opens a significant gap, while the remaining algorithms conclude the process at relatively similar performance levels.

The results for the third scenario are presented in Table 3 and Figures 6 and 7. This scenario is defined by a large population size combined with a limited number of iterations. Such a combination naturally provides diversity to the algorithms but does not offer sufficient time to enhance solution quality. In this context, unlike previous scenarios, population-induced diversity ceases to be an advantage, and elitist strategies capable of acting quickly in the early stages come to the forefront.

This shift is most dramatically reflected in the performance of GA06. Although GA06 previously stood out due to its structural support for diversity, it fails to maintain that advantage under this scenario. On the contrary, the continued application of positive discrimination toward weak individuals and the limited selection of strong ones in a large population setting effectively turns its former advantage into a disadvantage. The decline in performance that began in the second scenario continues here as well, with GA06 falling even behind GA02.

On the other hand, GA05 and GA03, which had previously appeared relatively disadvantaged due to short evolutionary durations, are now able to realize their potential thanks to the increased population size. GA05 once again achieves the lowest Median value, as it did in the second scenario. The fact that this value is nearly identical to that of the previous scenario suggests that for GA05, there is little difference between having a large population versus a high iteration count.

GA03 follows closely behind GA05, and the performance gap between them appears to have narrowed considerably compared to the previous scenario. This indicates that GA03 benefits slightly more from large population settings.

The convergence curves preserve the overall structure observed earlier. However, a notable observation is that GA03 remains neck-and-neck with GA05 until almost the end of the process. In the final few iterations, GA05 gains a slight edge and finishes ahead.

The results for the fourth and final scenario are presented in Table 4 and Figures 8 and 9. This scenario, characterized by both a large population size and a high number of iterations, represents the most ideal evolutionary conditions. This combination enables the algorithms to explore a wide solution space while also applying effective exploitation through the selection of high-quality individuals over an extended period.

Therefore, under this scenario, the overall evolutionary performance of the algorithms is revealed in the clearest and most balanced manner.

The results clearly demonstrate the impact of this ideal setting. GA05 once again emerged as the most successful algorithm, achieving the lowest Median value across all scenarios. This is further supported by its boxplot, which exhibits a narrow structure with very few outliers, indicating that the algorithm produced results that were not only good but also highly consistent. Additionally, GA05 achieved the lowest Std value by a significant margin. This confirms that the elitist strategy yields maximum efficiency when combined with a large population and an extended evolutionary process.

Table 3 Optimization results for N=100 and T=20

Algorithms	Statistical Summary of Fitness Values			
	Best	**Median**	**Worst**	**Std**
GA01	5.7042	32.9837	97.1225	16.6028
GA02	4.9312	17.9718	61.2656	12.8229
GA03	4.5388	12.9656	81.0595	14.1293
GA04	4.8124	21.8115	70.5730	15.7836
GA05	4.4754	10.3718	107.9555	19.2538
GA06	5.7045	20.8530	72.1362	13.5063

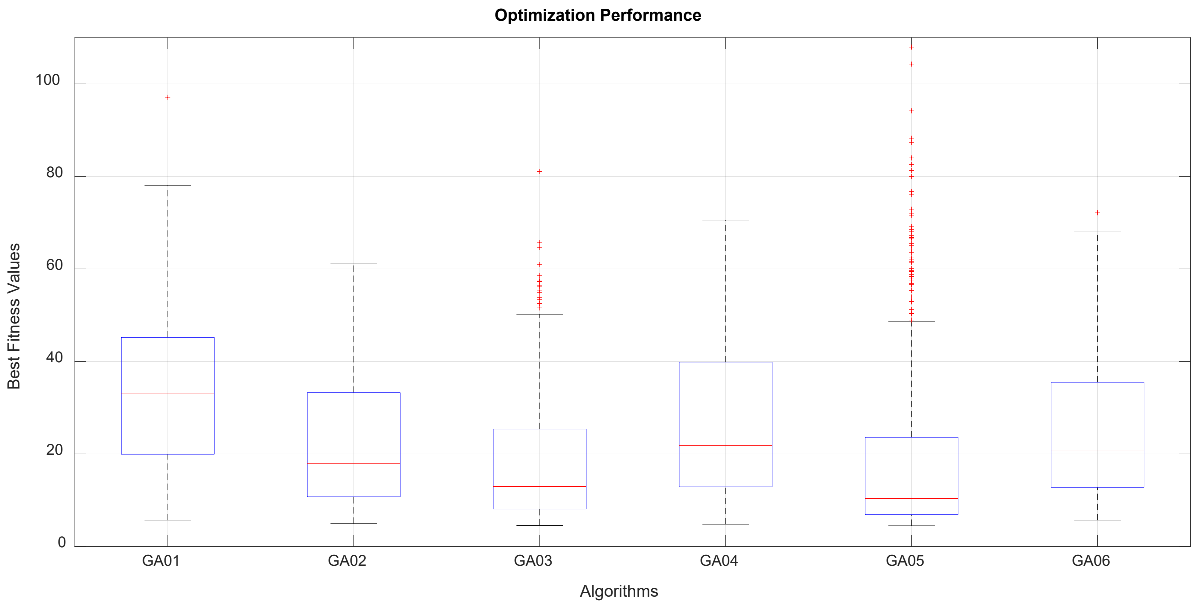
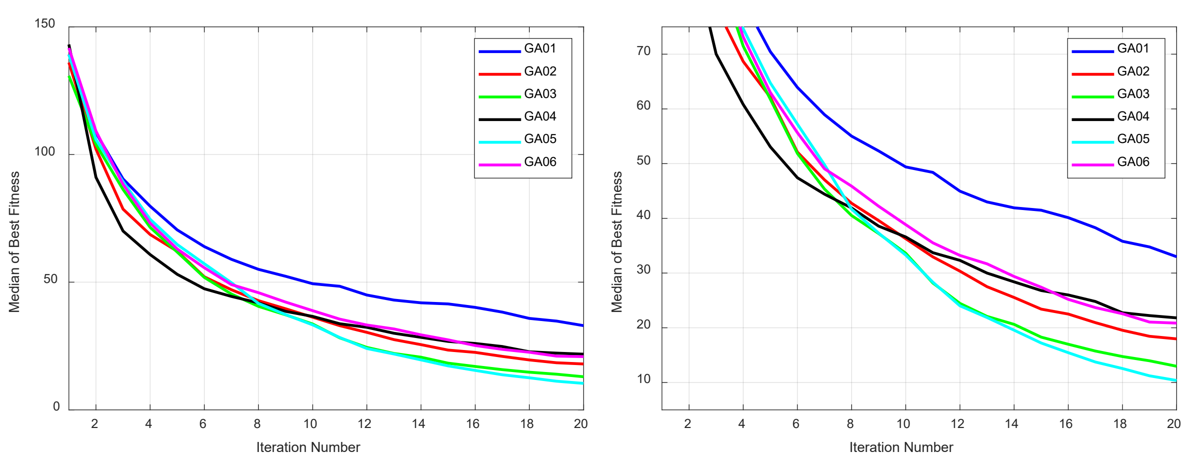


Figure 6 Boxplot graphic of the algorithms for N=100, T=20



(a) **(b)**

Figure 7 Convergence curves of the algorithms for N=100, T=20; **(a)** normal, **(b)** magnified view

Table 4 Optimization results for N=100 and T=100

Algorithms	Statistical Summary of Fitness Values			
	Best	**Median**	**Worst**	**Std**
GA01	4.6144	10.6678	44.5221	11.3218
GA02	4.3664	5.1320	41.8507	11.5785
GA03	4.3605	5.1953	44.0698	8.9830
GA04	4.4891	8.6848	42.8013	11.4411
GA05	4.3387	4.7604	13.9133	1.2981
GA06	4.5438	7.5069	44.6007	12.4250

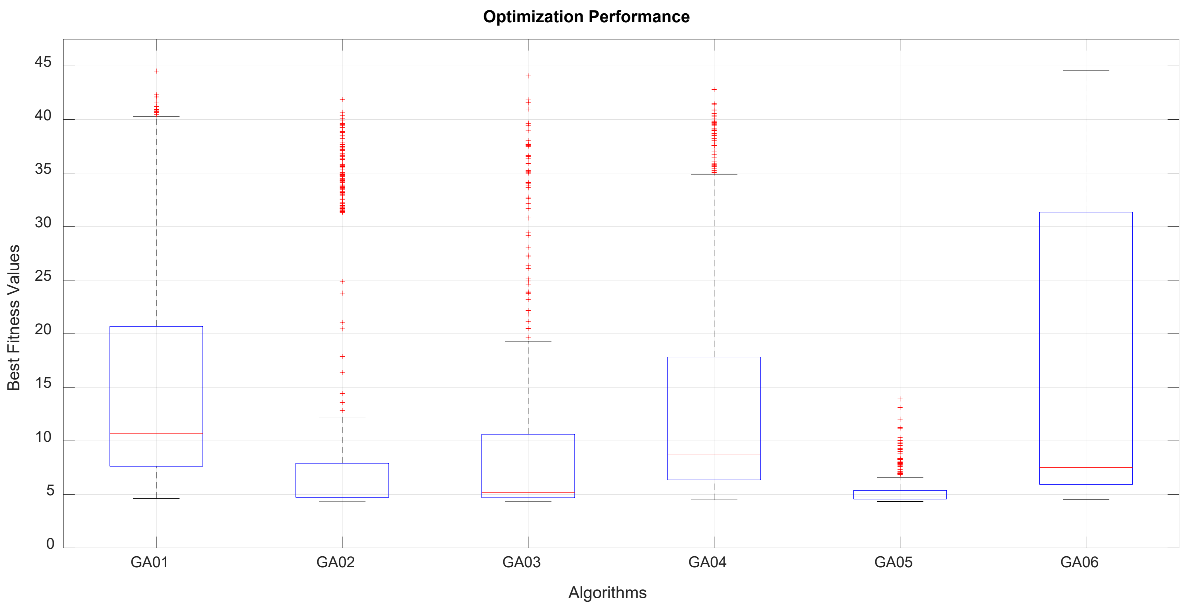


Figure 8 Boxplot graphic of the algorithms for N=100, T=100

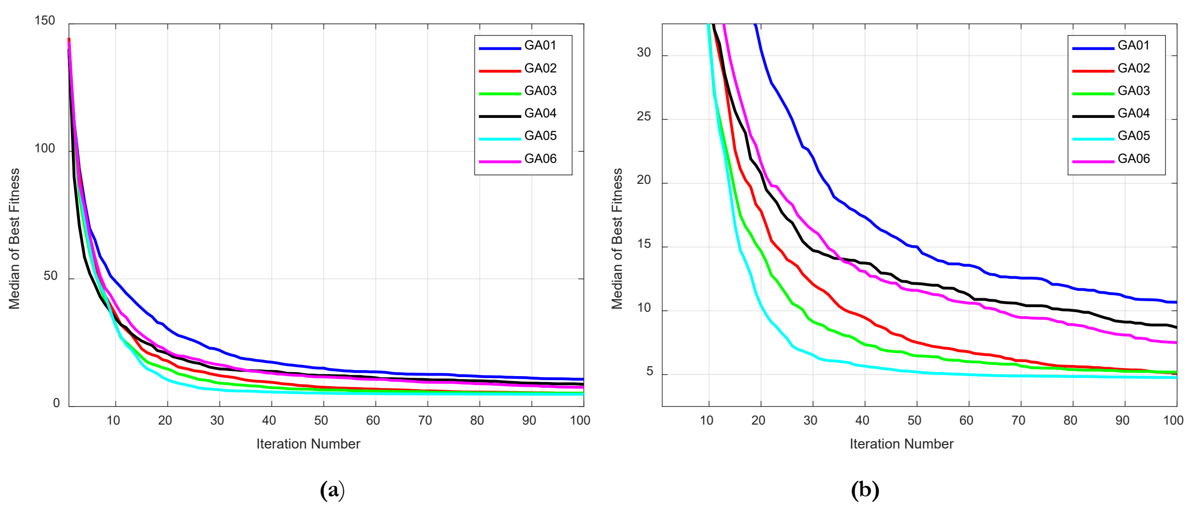


Figure 9 Convergence curves of the algorithms for N=100, T=100; **(a)** normal, **(b)** magnified view

GA02 managed to maintain its competitiveness over the long run, owing to its initial diversity advantage and balanced level of elitism. The performance of GA03 in this scenario also aligns well with previous trends. In terms of the Median metric, there is virtually no difference between GA02 and GA03, and the gap between either of them and GA05 remains relatively small.

However, the main distinction becomes apparent in the standard deviation (Std) values. The Std values of GA02 and GA03 are notably higher than that of GA05. Between the two, GA03 appears to have produced slightly more consistent results. Although the boxplot of GA02 may initially suggest a narrower spread, its high Std value and the concentration of outliers above the box indicate greater variability in its results.

As for GA06, the performance decline observed in previous scenarios continued. Even under this high-population condition, its structure—designed to protect weak individuals—caused it to lag behind. GA04 also underperformed, once again falling short of expectations and ranking among the worst-performing algorithms along with GA01.

The convergence curves clearly reflect the overall trend. GA05, GA03, and GA02 distinctly separate themselves from the other algorithms throughout the process. Starting from the early iterations, GA05 demonstrated the fastest decline in fitness, reaching low values in a short time. However, midway through the process, the rate of decline slowed, and it transitioned into a more stable convergence pattern. Meanwhile, GA03 and GA02, though initially declining more gradually than GA05, consistently improved in later iterations, and by the end, all three algorithms converged to similarly high-quality solutions.

To evaluate the optimization results at the waveform level, the switching angles corresponding to the Median metric for each algorithm and scenario were applied to a CHB-MLI model developed in MATLAB/Simulink. The output voltage waveforms were analyzed by extracting their harmonic spectra, which included the amplitudes of the fundamental component and harmonics up to the 10th order, along with THD levels. The corresponding simulation results are presented in Tables 5–8 and Figures 10–13.

As observed in the results, the THD levels and fundamental component errors are generally consistent with the previously reported fitness values, since these targets were explicitly incorporated into the fitness function. These additional simulation findings visually confirm the quality of the obtained solutions and provide further insights into the waveform-level behavior of the algorithms.

As expected, in the first scenario—characterized by limited population size and low iteration count—the overall performance of the algorithms remained constrained, with none of them achieving a THD value below 1%. Moreover, in some algorithms, a trade-off between the two optimization targets was observed; while THD was reduced, the voltage error increased, or vice versa. Nevertheless, GA02 and GA06, which prioritize diversity, outperformed the others under these constraints by achieving lower THD and voltage error levels, highlighting the advantage of diversity-oriented structures in resource-constrained environments.

In contrast, the fourth scenario, with its large population size and extended evolutionary duration, led to improved overall performance across all algorithms. All achieved THD values below 1%, and the highest voltage error remained within 1.6%. Despite this general improvement, GA02, GA03, and GA05—each employing strong and controlled elitist strategies—distinctly outperformed the remaining algorithms in terms of both harmonic suppression and fundamental component accuracy

Table 5 Optimal switching angles and simulation results for Median metric (N=20,T=20)

Algorithms	Fitness Scores	Optimal Switching Angles (θ) θ_1	θ_2	θ_3	θ_4	θ_5	% THD Until 10th	% Error V_1 Voltage
GA01	85.3245	7.9465	18.3534	29.7735	45.8827	65.4568	2.93	-0.32
GA02	53.5927	1.5109	23.3400	36.4685	60.9955	89.8899	1.03	-18.36
GA03	59.8713	11.6788	15.9557	35.1780	53.7048	83.7996	1.90	-12.04
GA04	75.7216	10.0911	15.0053	34.0250	51.6044	83.9939	2.33	-10.84
GA05	88.4709	3.1617	18.7100	29.9396	41.0352	67.7889	3.11	0.36
GA06	48.1857	4.9903	20.2231	33.9298	53.5520	85.7324	1.52	-12.68

Table 6 Optimal switching angles and simulation results for Median metric (N=20,T=100)

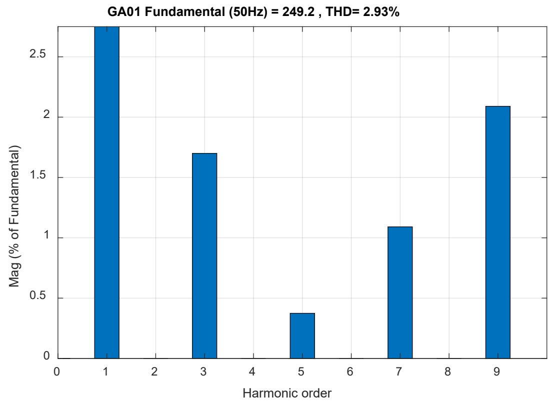
Algorithms	Fitness Scores	Optimal Switching Angles (θ) θ_1	θ_2	θ_3	θ_4	θ_5	% THD Until 10th	% Error V_1 Voltage
GA01	35.2412	1.4836	20.7879	27.1360	45.4366	65.2832	1.91	0.36
GA02	15.2062	8.7132	13.4753	32.6415	41.8628	65.4789	1.26	0.8
GA03	14.8900	1.2217	20.2579	28.1440	42.7254	64.3385	1.10	1.44
GA04	30.5692	9.4952	13.0744	36.6930	37.5135	64.5422	1.92	1.4
GA05	10.1239	3.0639	18.1193	29.3707	43.7832	64.5049	0.77	1.08
GA06	16.4486	3.9476	17.5527	30.5590	41.3452	65.1237	1.20	1.36

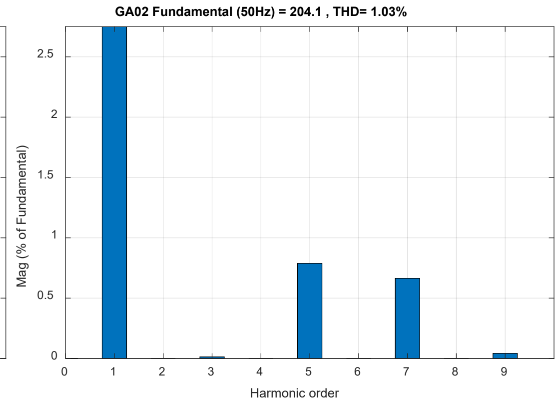
Table 7 Optimal switching angles and simulation results for Median metric (N=100,T=20)

Algorithms	Fitness Scores	Optimal Switching Angles (θ) θ_1	θ_2	θ_3	θ_4	θ_5	% THD Until 10th	% Error V_1 Voltage
GA01	32.9837	0.6173	19.2967	29.2596	45.5644	65.6026	1.56	-0.04
GA02	17.9718	9.2698	14.1192	32.0693	43.3065	66.3290	1.21	0.08
GA03	12.9656	4.1939	17.6740	28.6875	42.8704	63.5129	0.97	1.92
GA04	21.8115	5.6635	16.4219	31.2042	41.7955	61.7783	1.16	2.48
GA05	10.3718	10.0673	13.3050	31.9323	42.9492	65.3629	0.80	0.6
GA06	20.8530	3.0644	17.1574	29.8408	40.7698	61.9660	0.94	3

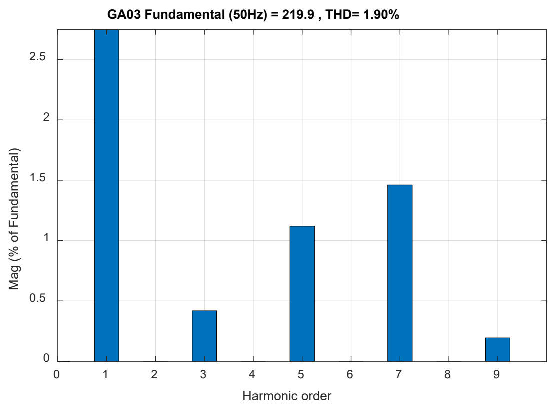
Table 8 Optimal switching angles and simulation results for Median metric (N=100,T=100)

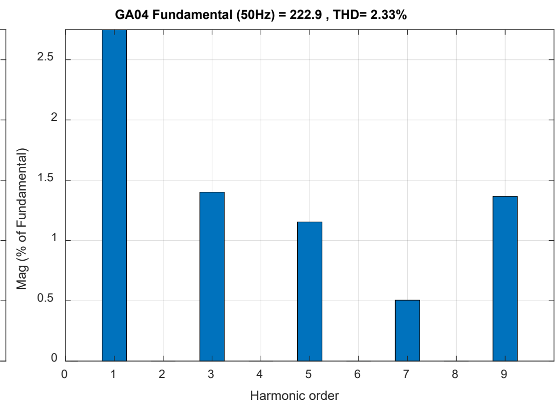
Algorithms	Fitness Scores	Optimal Switching Angles (θ) θ_1	θ_2	θ_3	θ_4	θ_5	% THD Until 10th	% Error V_1 Voltage
GA01	10.6678	0.0314	19.4255	28.5023	42.4662	64.3338	0.81	1.56
GA02	5.1320	0.3848	18.9488	29.4598	42.7833	63.6613	0.23	1.6
GA03	5.1953	0.0580	19.0705	29.0255	43.0818	63.8637	0.33	1.52
GA04	8.6848	2.0110	18.2852	29.1365	42.6399	62.6593	0.59	2.2
GA05	4.7604	4.1916	17.6921	30.1474	42.6098	63.9630	0.23	1.48
GA06	7.5069	4.7167	17.2323	30.8623	42.8791	64.6001	0.61	1.04



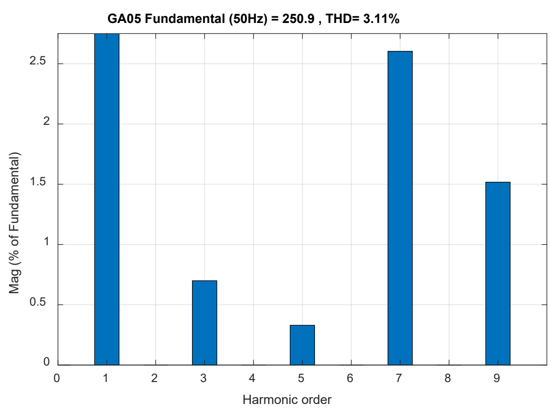


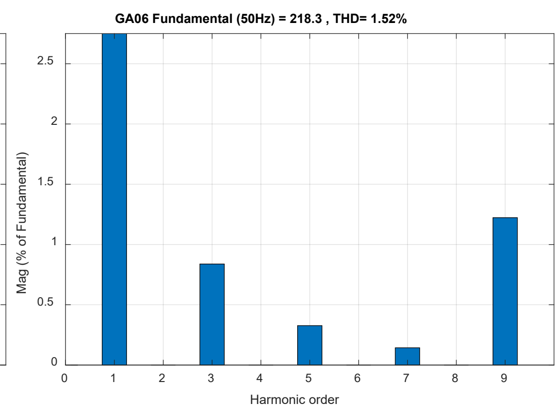
(a) (b)





(c) (d)





(e) (f)

Figure 10 Frequency spectrum and fundamental harmonic values of algorithms at N=20, T=20

(a) GA01; **(b)** GA02; **(c)** GA03; **(d)** GA04; **(e)** GA05; **(f)** GA06

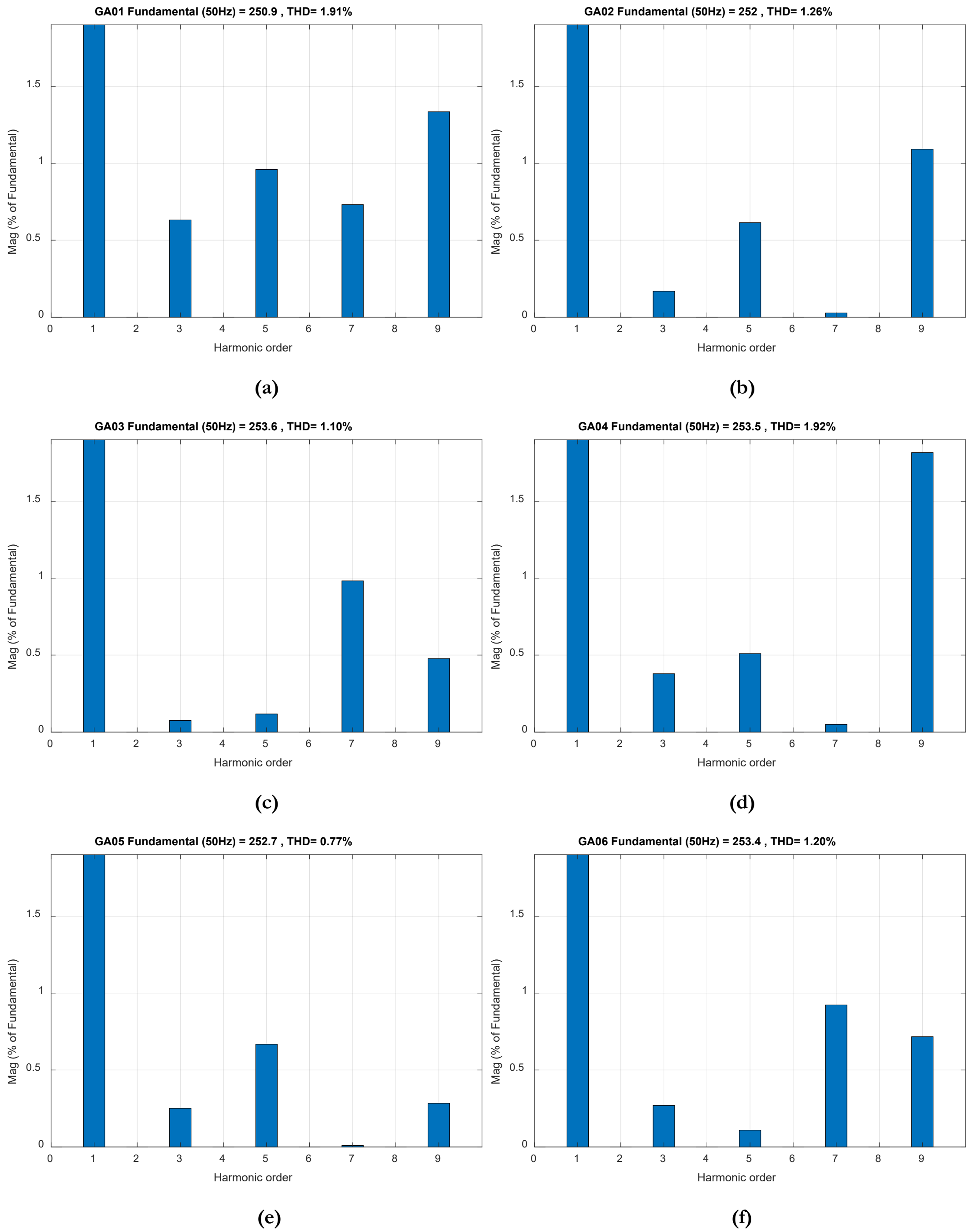


Figure 11 Frequency spectrum and fundamental harmonic values of algorithms at N=20, T=100

(a) GA01; **(b)** GA02; **(c)** GA03; **(d)** GA04; **(e)** GA05; **(f)** GA06

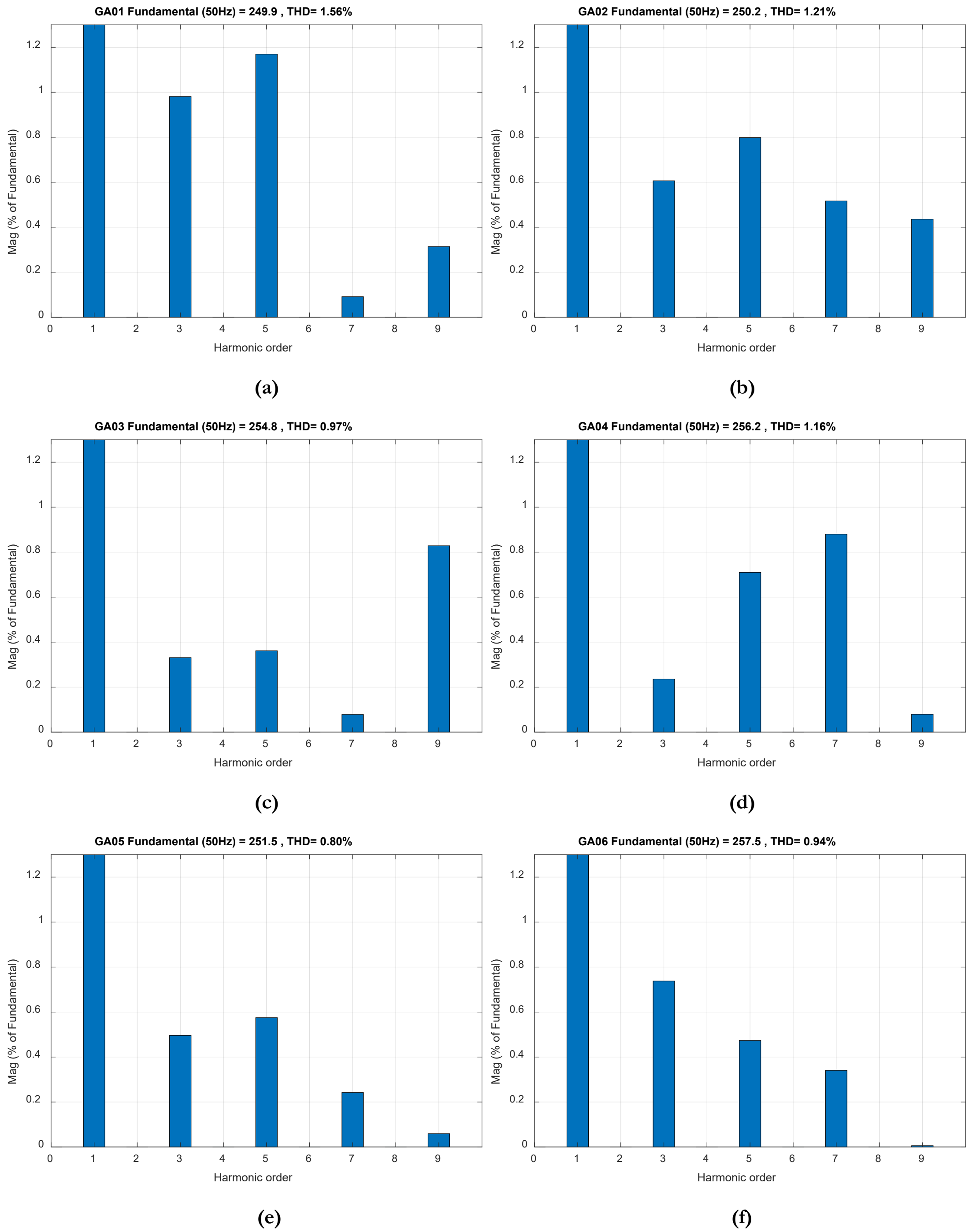


Figure 12 Frequency spectrum and fundamental harmonic values of algorithms at N=100, T=20

(a) GA01; **(b)** GA02; **(c)** GA03; **(d)** GA04; **(e)** GA05; **(f)** GA06

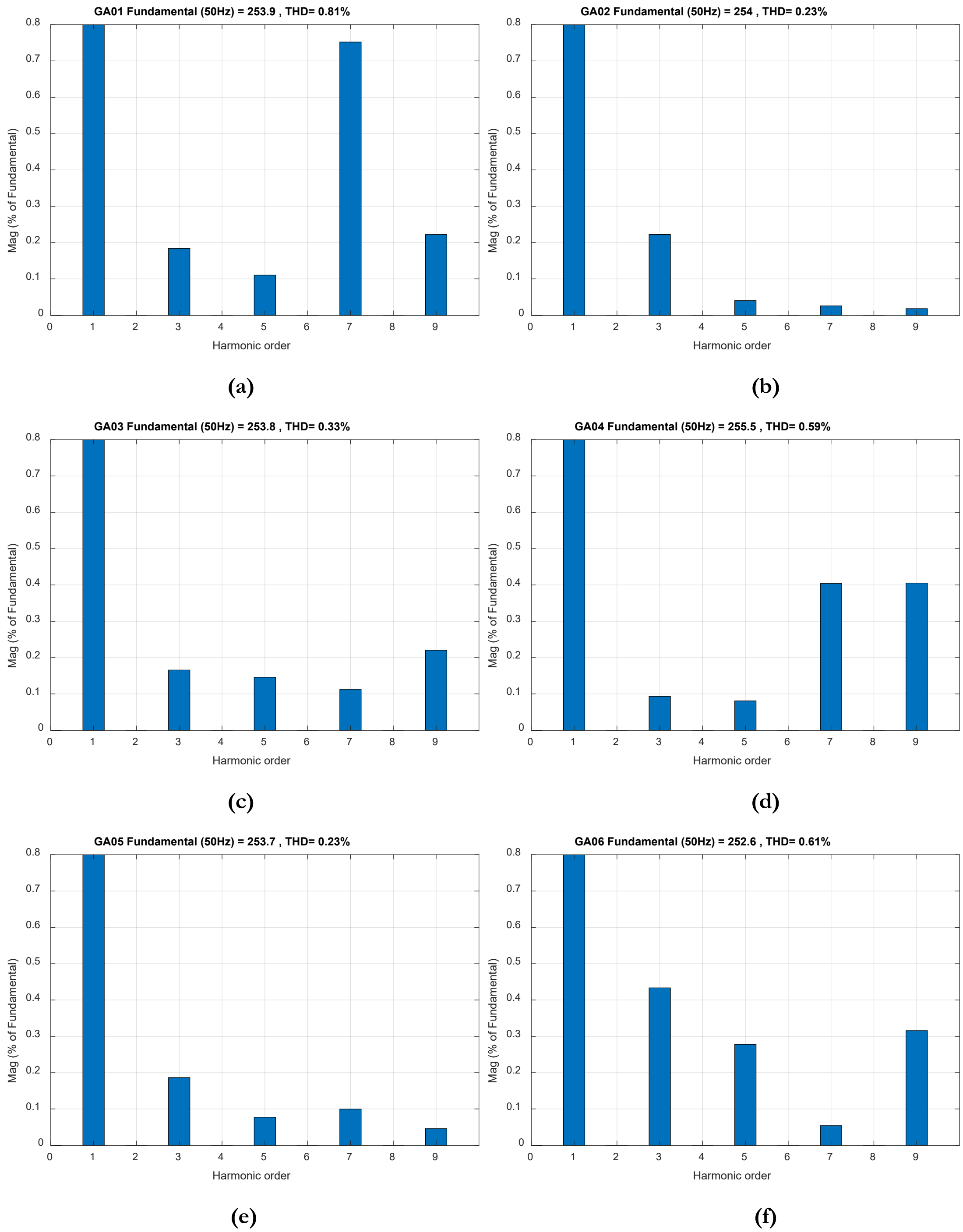


Figure 13 Frequency spectrum and fundamental harmonic values of algorithms at N=100, T=100

(a) GA01; **(b)** GA02; **(c)** GA03; **(d)** GA04; **(e)** GA05; **(f)** GA06

5. Conclusions

In this study, six genetic algorithm variants employing different generation replacement strategies were evaluated in terms of the balance between diversity and elitism and comparatively analyzed on a real-world optimization problem. The results obtained under four scenarios representing different resource conditions revealed that the performance of the algorithms depends not only on their structural design but also significantly on the resource profile of the environment in which they are applied. While strategies with strong elitism demonstrated superior performance in scenarios with high population sizes and iteration counts, diversity-oriented approaches stood out particularly in resource-constrained settings. These findings emphasize that, rather than seeking a universally superior algorithm, selection should be based on the principle of suitability to the specific problem and available resources.

These results offer a concrete roadmap for identifying which type of generation replacement strategy may be more effective under which resource conditions in real-world optimization problems. This evaluation approach can potentially be extended beyond genetic algorithms and adapted to other metaheuristic algorithms. In particular, conducting similar comparative analyses on multi-objective or constrained optimization problems may further strengthen the scope and validity of the insights obtained.

Declaration of Ethical Standards

As the sole author of this study, I confirm that the research was conducted in full compliance with all ethical standards.

Declaration of Competing Interest

As the sole author of this study, I declare that I have no conflict of interest.

Funding / Acknowledgements

As the sole author, I declare that this study received no external funding.

Data Availability

No datasets were generated or analyzed during the current study.

References

Ahn, C. W., & Ramakrishna, R. S. (2003). Elitism-based compact genetic algorithms. *IEEE Transactions on Evolutionary Computation*, *7*(4), 367–385. https://doi.org/10.1109/TEVC.2003.814633

Alhijawi, B., & Awajan, A. (2024). Genetic algorithms: Theory, genetic operators, solutions, and applications. *Evolutionary Intelligence*, *17*(3), 1245–1256. https://doi.org/10.1007/s12065-023-00822-6

Ali, K. B., Telmoudi, A. J., & Gattoufi, S. (2020). Improved Genetic Algorithm Approach Based on New Virtual Crossover Operators for Dynamic Job Shop Scheduling. *IEEE Access*, *8*, 213318–213329. https://doi.org/10.1109/ACCESS.2020.3040345

Bakirtzis, A. G., Biskas, P. N., Zoumas, C. E., & Petridis, V. (2002). Optimal power flow by enhanced genetic algorithm. *IEEE Transactions on Power Systems*, *17*(2), 229–236. https://doi.org/10.1109/TPWRS.2002.1007886

Bektaş, Y., Karaca, H., Taha, T. A., & Zaynal, H. I. (2023). Red deer algorithm-based selective harmonic elimination technique for multilevel inverters. *Bulletin of Electrical Engineering and Informatics*, *12*(5), 2643–2650. https://doi.org/10.11591/eei.v12i5.5160

Bertin, T., Despesse, G., & Thomas, R. (2023). Comparison between a Cascaded H-Bridge and a Conventional H-Bridge for a 5-kW Grid-Tied Solar Inverter. *Electronics*, *12*(8), 1–25. https://doi.org/10.3390/electronics12081929

Brouwer, N., Dijkzeul, D., Koppenhol, L., Pijning, I., & van den Berg, D. (2022). Survivor selection in a crossoverless evolutionary algorithm. *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 1631–1639. https://doi.org/10.1145/3520304.3533950

Dahidah, M. S. A., Konstantinou, G., & Agelidis, V. G. (2015). A Review of Multilevel Selective Harmonic Elimination PWM: Formulations, Solving Algorithms, Implementation and Applications. *IEEE Transactions on Power Electronics*, *30*(8), 4091–4106. IEEE Transactions on Power Electronics. https://doi.org/10.1109/TPEL.2014.2355226

Disney, S. M., Naim, M. M., & Towill, D. R. (2000). Genetic algorithm optimisation of a class of inventory control systems. *International Journal of Production Economics*, *68*(3), 259–278. https://doi.org/10.1016/S0925-5273(99)00101-2

Du, H., Wang, Z., Zhan, W., & Guo, J. (2018). Elitism and Distance Strategy for Selection of Evolutionary Algorithms. *IEEE Access*, *6*, 44531–44541. https://doi.org/10.1109/ACCESS.2018.2861760

El-Hosainy, A., Hamed, H. A., Azazi, Haitham. Z., & El-Kholy, E. E. (2017). A review of multilevel inverter topologies, control techniques, and applications. *2017 Nineteenth International Middle East Power Systems Conference (MEPCON)*, 1265–1275. https://doi.org/10.1109/MEPCON.2017.8301344

Gen, M., & Cheng, R. (1999). *Genetic Algorithms and Engineering Optimization.* John Wiley & Sons, Ltd. https://doi.org/10.1002/9780470172261

Halim, Z., Yousaf, M. N., Waqas, M., Sulaiman, M., Abbas, G., Hussain, M., Ahmad, I., & Hanif, M. (2021). An effective genetic algorithm-based feature selection method for intrusion detection systems. *Computers & Security*, *110*, 102448. https://doi.org/10.1016/j.cose.2021.102448

Hassanat, A., Almohammadi, K., Alkafaween, E., Abunawas, E., Hammouri, A., & Prasath, V. B. S. (2019). Choosing Mutation and Crossover Ratios for Genetic Algorithms—A Review with a New Dynamic Approach. *Information*, *10*(12), 1–36. https://doi.org/10.3390/info10120390

Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence* (5th ed.). The MIT Press. https://doi.org/10.7551/mitpress/1090.001.0001

Hong, T.-P., Wang, H.-S., Lin, W.-Y., & Lee, W.-Y. (2002). Evolution of Appropriate Crossover and Mutation Operators in a Genetic Process. *Applied Intelligence*, *16*(1), 7–17. https://doi.org/10.1023/A:1012815625611

Katoch, S., Chauhan, S. S., & Kumar, V. (2021). A review on genetic algorithm: Past, present, and future. *Multimedia Tools and Applications*, *80*(5), 8091–8126. https://doi.org/10.1007/s11042-020-10139-6

Kim, K., & Han, I. (2000). Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. *Expert Systems with Applications*, *19*(2), 125–132. https://doi.org/10.1016/S0957-4174(00)00027-0

Lingom, P. M., Song-Manguelle, J., Nyobe-Yome, J. M., & Doumbia, M. L. (2024). A Comprehensive Review of Compensation Control Techniques Suitable for Cascaded H-Bridge Multilevel Inverter Operation with Unequal DC Sources or Faulty Cells. *Energies*, *17*(3), 1–17. https://doi.org/10.3390/en17030722

Osman, M. S., Abo-Sinna, M. A., & Mousa, A. A. (2004). A solution to the optimal power flow using genetic algorithm. *Applied Mathematics and Computation*, *155*(2), 391–405. https://doi.org/10.1016/S0096-3003(03)00785-9

Smith, M. G., & Bull, L. (2005). Genetic Programming with a Genetic Algorithm for Feature Construction and Selection. *Genetic Programming and Evolvable Machines*, *6*(3), 265–281. https://doi.org/10.1007/s10710-005-2988-7

Sunddararaj, S. P., Srinivasarangan, S. R., & Nallusamy, S. (2020). An Extensive Review of Multilevel Inverters Based on Their Multifaceted Structural Configuration, Triggering Methods and Applications. *Electronics*, *9*(3), 1–33. https://doi.org/10.3390/electronics9030433

Yang, K., Zhang, Q., Zhang, J., Yuan, R., Guan, Q., Yu, W., & Wang, J. (2017). Unified Selective Harmonic Elimination for Multilevel Converters. *IEEE Transactions on Power Electronics*, *32*(2), 1579–1590. IEEE Transactions on Power Electronics. https://doi.org/10.1109/TPEL.2016.2548080

Zhang, F., Cao, X., & Yang, D. (2008). Intelligent scheduling of public traffic vehicles based on a hybrid genetic algorithm. *Tsinghua Science and Technology*, *13*(5), 625–631. https://doi.org/10.1016/S1007-0214(08)70103-2

Zhang, J., Zhuang, J., Du, H., & Wang, S. (2009). Self-organizing genetic algorithm based tuning of PID controllers. *Information Sciences*, *179*(7), 1007–1018. https://doi.org/10.1016/j.ins.2008.11.038