Optimizing Classic GA Performance Through Grey Code and Adaptive Techniques

Buzdea Stefan
Pricop Tudor-Andrei

December 5, 2023

1 Abstract

This paper investigates the enhanced efficiency of optimized genetic algorithms in minimizing benchmark functions, comparing their performance with the classical genetic algorithm. We explore various optimization techniques, including gray codification and adaptive operators, alongside an improved selection method—elitism—compared to Holland's roulette. Focusing on global minimum optimization across 5, 10, and 30 dimensions, we analyze key statistics such as average and best values, tracking the evolution of function minima across generations.

Our findings highlight the superior results of Gray codification, particularly in Rastrigin and Michalewicz functions, due to their smaller solution spaces within reasonable timeframes. Additionally, implementing elitism selection and the adaptive Hypermutation operator successfully optimizes every function, with increased mutation probability for significant improvements. Notably, in higher dimensions, optimized genetic algorithms demonstrate heightened accuracy.

In conclusion, this study emphasizes the advantages of optimization methods in genetic algorithms for diverse benchmark functions.

2 Introduction

This study endeavors to compare classical Genetic Algorithms with optimized counterparts in various scenarios. Genetic algorithms simulate natural selection, favoring the reproduction of the fittest individuals across generations. Utilizing biologically inspired operators like mutation, crossover, and selection, these algorithms excel in solving optimization problems. The traditional approach follows a standard five-step process, while the optimized version leverages specific insights, enabling the generation of superior solutions.

Our investigation focuses on four benchmark functions—De Jong 1, Schwefel's, Rastrigin's, and Michalewicz's—across dimensions of 5, 10, and 30, implemented in Python. Through meticulous experimentation, we aim to extract minimum values, average minimum values, and execution times for each method. By observing the evolution of function minima over generations, we seek to discern notable differences, providing nuanced insights into the efficacy of each optimization method in various scenarios.

This research not only aims to pinpoint the most effective optimization method for global minima discovery but also delves into practical implications.

In conclusion, we will succinctly summarize key insights, elucidating their relevance in the realm of optimization problems.

3 Methods

First we will describe the classical genetic algorithm and then the optimization methods we thought of and why.

Classical Genetic Algorithm:

In the genetic algorithm implementation, the entire population is represented as a vector of bit vectors generated randomly. We determined the required number of bits to represent each interval, considering the function's dimensionality, and maintained a precision of 10^5, consistent with the approach used in the previously discussed methods. After generating a population of size *pop_size* chromosomes and calculating the function values at those points, the population evolution process begins over a set number of generations. During each generation, a population of chromosomes is selected, followed by the application of the mutation and crossover operators. Holland's roulette wheel selection is employed for selection, allowing individuals to be chosen in the next population generation with a probability proportional to their fitness divided by the population's total fitness.

The fitness function, designed to measure chromosomal quality and ensure that superior candidates have a higher probability of selection, is computed using the formula:

$$f(x) = \left(\frac{e_{\min} - e(x)}{e_{\max} - e_{\min} + \epsilon} + 1 \right)^{\text{selection pressure}}$$

Here, ϵ is set to 10^{-6} to minimize its impact on the fraction.

For the mutation process, a random number between $[0, 1]$ is chosen for each bit of each individual. If it is less than $1/L$ (where L is the number of bits in a chromosome), the bit is mutated, resulting in a flip. In the crossover function, each individual is assigned a random number in $[0, 1]$. Individuals are then ordered based on these numbers, and pairs are formed only if their assigned number is less than 0.8, indicating an 80% chance of generating new descendants.

Two randomly selected cutting points are employed for the crossover process. A noteworthy optimization involves a greedy method, where, from the two parents and resulting two descendants, only the best two are chosen. This adjustment, prompted by suboptimal outcomes in strategies where descendants replaced parents or all four individuals were retained, ensures a more effective genetic algorithm by prioritizing the most promising individuals. At the conclusion of all iterations through the generations, the best chromosome from the last generation is selected as the solution.

This methodology balances computational efficiency and optimization effectiveness within the genetic algorithm framework.

Optimized Genetic Algorithm:

In refining the genetic algorithm for function minimization, several enhancements were incorporated to augment performance across diverse functions. The first pivotal modification introduced is the integration of elitism. By preserving the best k individuals in each generation, irrespective of the selection outcome, we ensure a continuous presence of high-performing candidates in subsequent populations. This strategic inclusion mitigates the risk of losing exceptional solutions due to random selection or the application of different genetic operators.

The second notable refinement involves the adoption of Gray Codification. Departing from the conventional approach of generating random bitstrings and decoding them as binary representations, Gray Codification is employed. This optimization, particularly impactful for Rastrigin and Michalewicz's functions with smaller solution spaces, contributes to improved outcomes. However, it is observed that for other functions, there is no discernible enhancement, and in some cases, a slight degradation in solution quality is noted.

The third optimization strategy involves incorporating an adaptive operator: Hypermutation. This dynamic operator assesses the evolutionary trajectory of the best individual in the current generation in comparison to its state n generations ago. Essentially, this informs us that if the population converges too rapidly, it is advisable to increase the mutation probability. We have categorized this into two scenarios: initially, for the first 1500 generations, and subsequently, for the last 500 generations. Our observations indicate that, initially, it is preferable not to elevate the mutation probability unless the observed difference is substantial. Towards the end, when the solution tends to converge more quickly, and the results are closely clustered, a higher mutation probability is warranted if the observed difference changes less compared to the initial period.

This methodology ensures diversity in the population proportional to the current generation number.

4 Functions

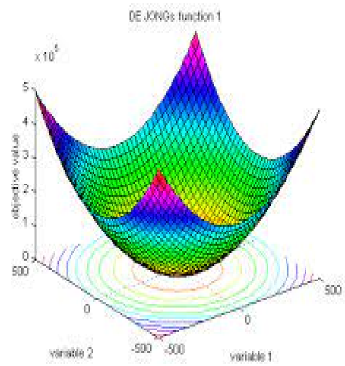


Figure 1: De Jong's function: $f_n(x) = \sum_{i=1}^{n} ix_i^2$
$-5.12 \leq x_i \leq -5.12$
global minimum: $f(x) = 0$

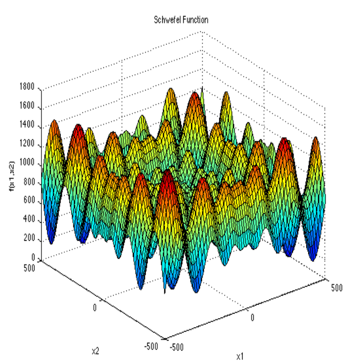


Figure 2: Schwefel's function: $f_n(x) = \sum_{i=1}^{n} -x_i sin(\sqrt{|x_i|})$
$-500 \leq x_i \leq 500$
global minimum: $f_n(x) = -n * 418.9829$

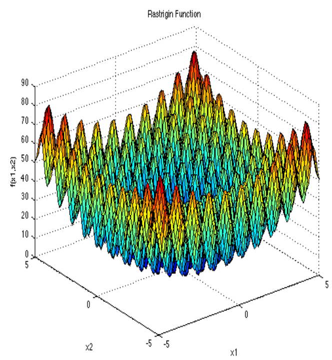


Figure 3: Rastrigin's function: $f_n(x) = 10n + \sum_{i=1}^{n}(x_i^2 - 10cos(2\pi x_i))$
$-5.12 \leq x_i \leq -5.12$
global minimum: $f(x) = 0$

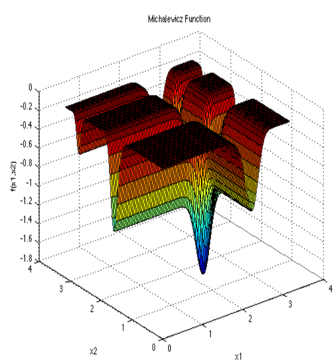


Figure 4: Michalewicz's function: $f_n(x) = -\sum_{i=1}^{n} sin(x_i)(sin\frac{ix_i^2}{\pi})^{20}$
$0 \leq x_i \leq \pi$
global minimum: $f_5(x) = -4.687, f_{10}(x) = -9.66$

5 Experimental results

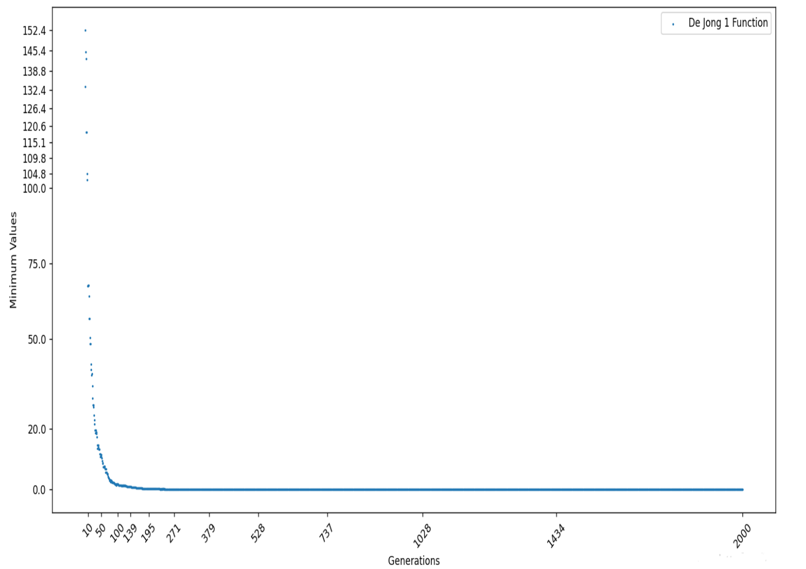
Must be noticed that OGA stands for Optimized Genetic Algorithm.

DeJong				
Dimension	Algorithm	Minim	Average	Time
30	GA	0.0024	0.0069	184 s
	OGA	0.0019	0.0057	195 s
10	GA	0.0	0.0	121 s
	OGA	0.0	0.0	135 s
5	GA	0.0	0.0	80 s
	OGA	0.0	0.0	91 s

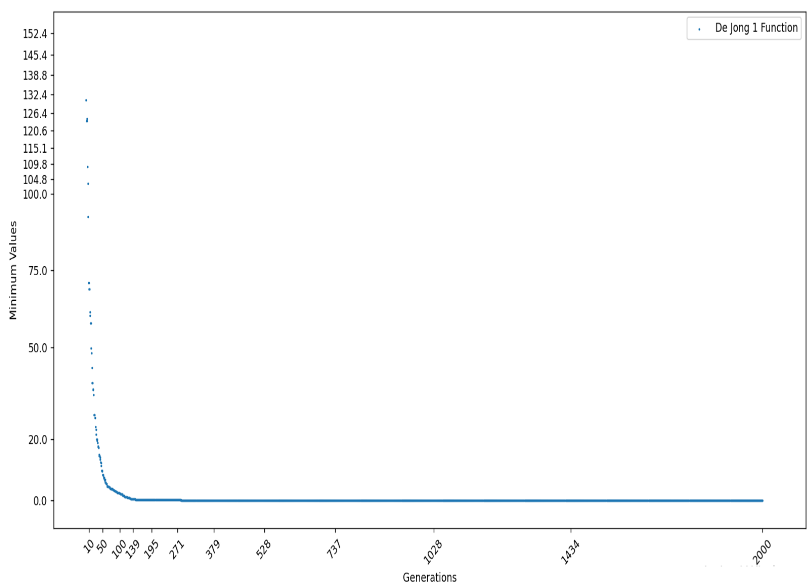
Rastrigin				
Dimension	Algorithm	Minim	Average	Time
30	GA	15.836	19.9776	195 s
	OGA	3.21028	8.20354	211 s
10	GA	0.0	1.18684	128 s
	OGA	0.0	0.90472	132 s
5	GA	0.0	0.04119	90 s
	OGA	0.0	0.03237	96 s

Schwefel				
Dimension	Algorithm	Minim	Average	Time
30	GA	-12533.25476	-12056.17361	179 s
	OGA	-12561.13219	-12527.18745	172 s
10	GA	-4189.72557	-4029.09442	114 s
	OGA	-4189.63942	-4102.54868	111 s
5	GA	-2094.91013	-2066.58240	94 s
	OGA	-2094.91421	-2086.78902	89 s

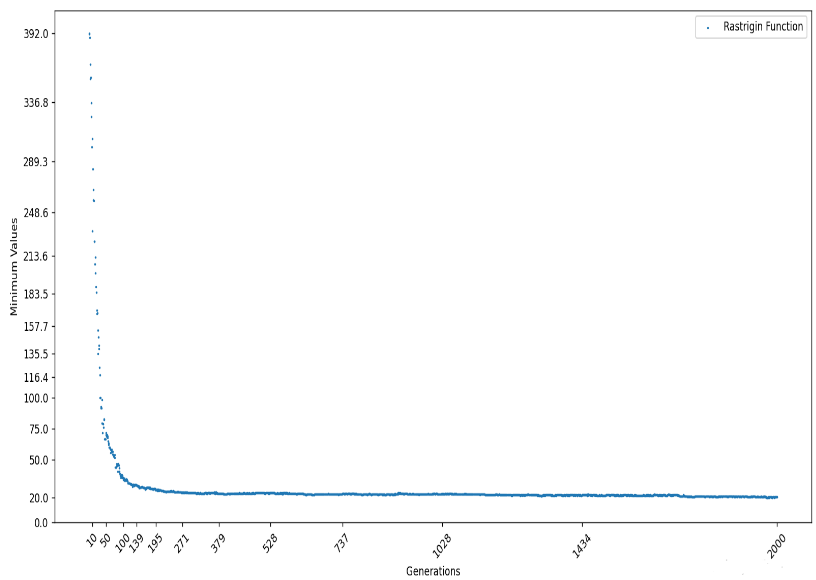
Michalewicz				
Dimension	Algorithm	Minim	Average	Time
30	GA	-28.5601	-28.1894	187 s
	OGA	-29.63822	-29.35803	193 s
10	GA	-9.6535	-9.6247	133 s
	OGA	-9.66003	-9.64947	137 s
5	GA	-4.68765	-4.6798	76 s
	OGA	-4.68765	-4.68702	85 s



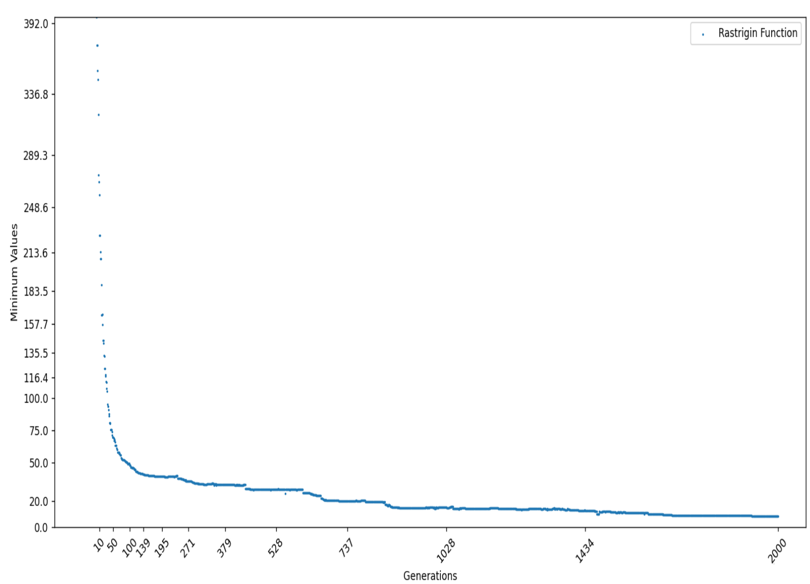
Genetic Algorithm Evolution on De Jong 1 function (30 dimensions)



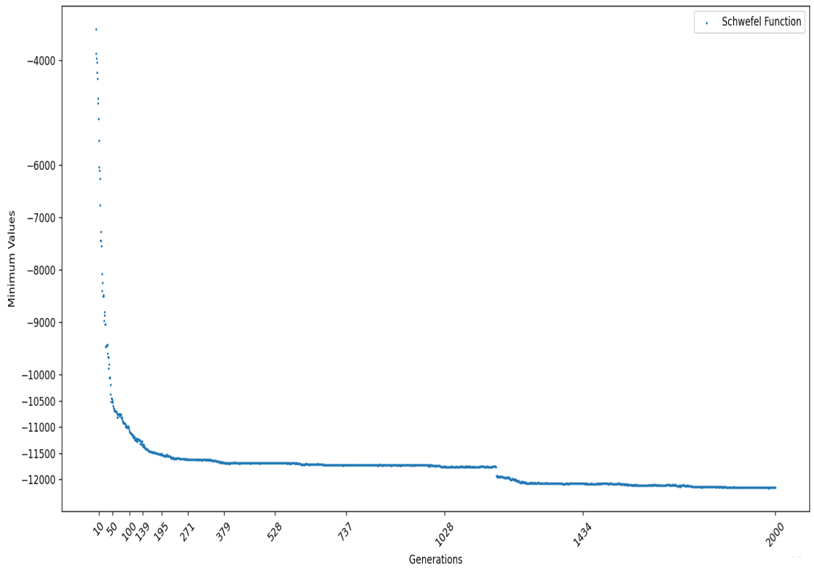
Optimized Genetic Algorithm Evolution on De Jong 1 function (30 dimensions)



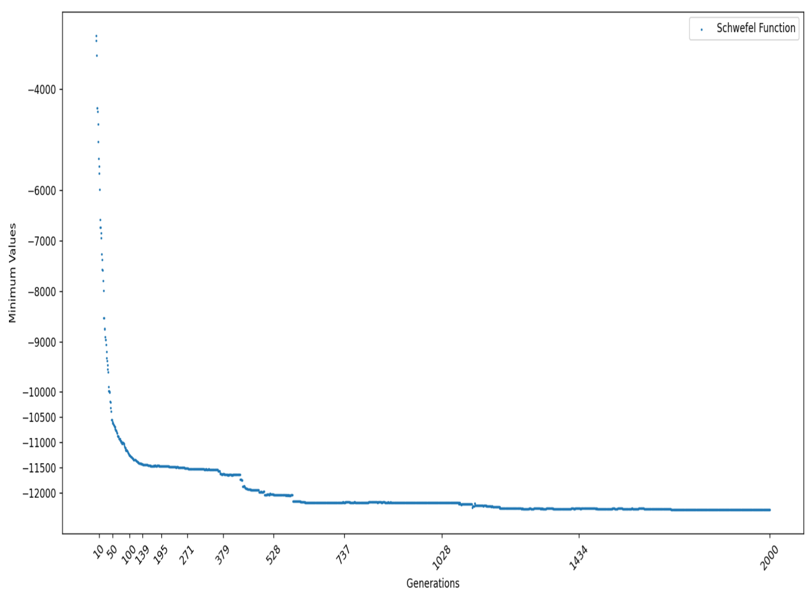
Genetic Algorithm Evolution on Rastrigin function (30 dimensions)



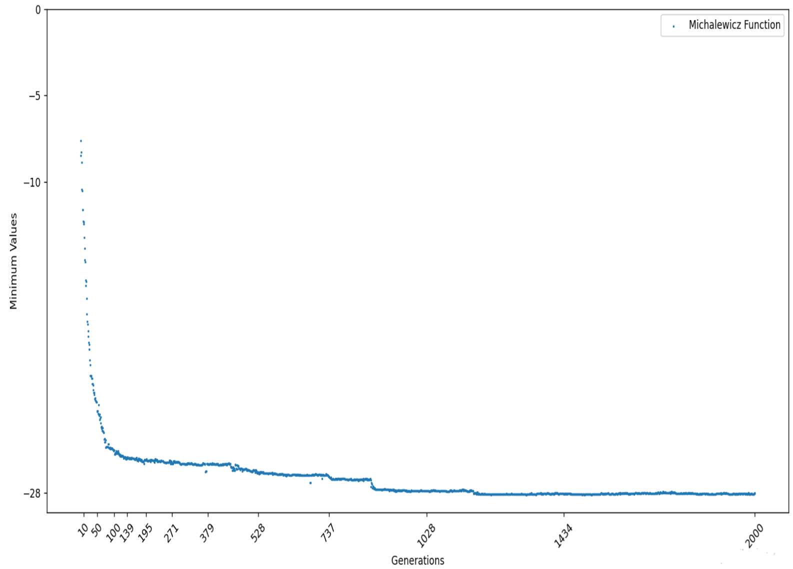
Optimized Genetic Algorithm Evolution on Rastrigin function (30 dimensions)



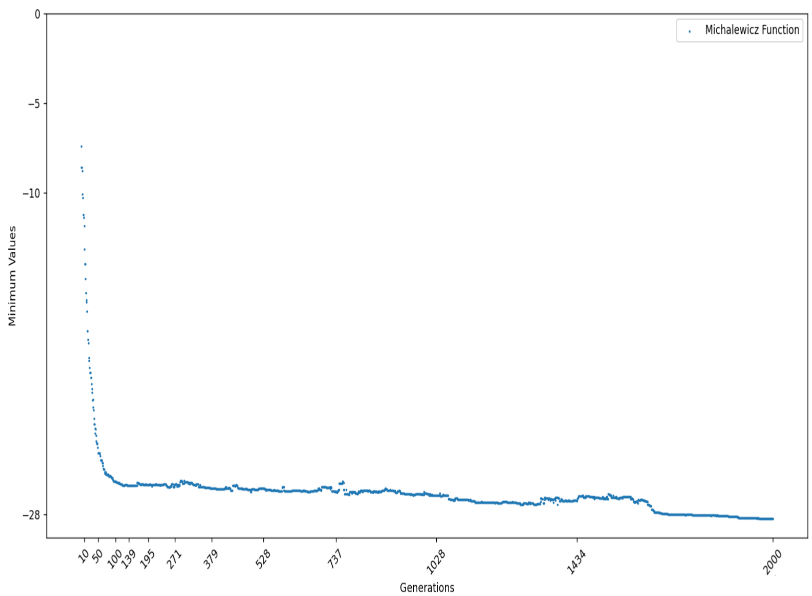
Genetic Algorithm Evolution on Schwefel function (30 dimensions)



Optimized Genetic Algorithm Evolution on Schwefel function (30 dimensions)



Genetic Algorithm Evolution on Michalewicz function (30 dimensions)



Optimized Genetic Algorithm Evolution on Michalewicz function (30 dimensions)

6 Experimental Description

In our experimental setup, we employed an Enhanced Genetic Algorithm to address the minimization problem across various functions with distinct dimensions. Our primary emphasis was on attaining precision in higher dimensions, ensuring results accurate to at least 5 decimal places. To validate the reliability of our outcomes, we conducted 30 independent runs for each dimension (5, 10, and 30), extracting key statistics such as minimum, average value, and average time. Throughout these runs, the population size (`pop_size`) and number of generations remained constant at 200 and 2000, respectively. The crossover probability was consistently set at 0.8, and elitism population selection involved the top k individuals, determined by a ratio of $k = 0.07 \times \texttt{pop\_size}$.

For the mutation probability, we implemented an adaptive approach, comparing the best individual from the current generation with the best individual from 20 (n) generations ago. Two distinct scenarios emerged based on the current generation number:

1. If the current generation number is ≤ 1500: If the absolute difference between the two individuals is less than the length of the function domain divided by 10, the mutation probability is adjusted to 3 times the normal mutation probability. The normal mutation probability is calculated as $1/L$, where L represents the number of bits in a chromosome.

2. If the current generation number is > 1500: If the absolute difference between the two individuals is less than the length of the function domain divided by 100, the mutation probability is set to 2 times the normal mutation probability, following the same calculation as mentioned above.

Accompanying graphical representations illustrate the convergence of each 30-dimensional function using the enhanced genetic algorithm. These visualizations depict the progressive enhancement of the best value at each generation over time, offering a comparative analysis with the classical genetic algorithm. The refined algorithm showcases improved adaptability, thereby augmenting its efficacy in exploring solution spaces across diverse dimensions.

7 Comparison and Interpretation

Upon scrutinizing the aforementioned results, the optimized Genetic Algorithm showcases outstanding performance, notably excelling in 30 dimensions and consistently surpassing classical Genetic Algorithm values, all while maintaining comparable execution times. Particularly in the case of the Rastrigin function with 30 dimensions, the optimization yields highly favorable results, affirming the success of our enhancements.

A noteworthy observation from the graphical representations is the increased variability in the best solution across generations in the optimized Genetic Algorithm, attributed to HyperMutation.

This contrasts with the classical algorithm, where a more restrained mutation strategy leads to faster overall convergence, albeit within a more limited solution space. Additionally, the incorporation of elitism ensures that the best individual in each generation outperforms its counterpart in the classical algorithm. This underscores the effectiveness of our optimization strategy in navigating diverse solution spaces.

8 Conclusion

In conclusion, our exploration of optimized genetic algorithms for function minimization has yielded notable advancements compared to the classical genetic algorithm. The integration of Gray Codification, elitism, and the adaptive Hypermutation operator significantly enhanced performance, with particularly commendable results in 30-dimensional scenarios. The Rastrigin function, in particular, showcased remarkable improvements, affirming the success of our optimization strategies.

The increased variability in the best solution across generations, facilitated by HyperMutation, introduces a dynamic element to the optimization process. While this contrasts with the classical algorithm's faster convergence within a more confined solution space, the trade-off allows the optimized algorithm to explore a broader range of potential solutions.

Furthermore, the incorporation of elitism ensures consistently superior individuals in each generation, contributing to the overall effectiveness of our optimization strategy. Notably, the optimized algorithm excelled in precision and accuracy, maintaining competitiveness in execution times.

In summary, our findings emphasize the advantages of employing optimization methods, including Gray Codification, elitism, and adaptive operators, in genetic algorithms. This study provides valuable insights into the nuanced dynamics of optimization strategies, highlighting their efficacy in navigating diverse solution spaces for a range of benchmark functions and dimensions.

References

[1] G. Bonori, M. Barbieri, F. Pellicano *Optimum profile modifications of spur gears by means of genetic algorithms.* ScienceDirect, 2008.

[2] F. Herrera, M. Lozano *Fuzzy adaptive genetic algorithms: design, taxonomy, and future directions.* Soft Computing, 2003.

[3] L. Budin, M. Golub, D. Jakobovic *Parallel Adaptive Genetic Algorithm*, 2023.

[4] S. Han, L. Xiao *An improved adaptive genetic algorithm.* SHS Web of Conferences, 2022.

[5] D. Bingham, S. Surjanovic *Test Functions and Datasets: Schwefel Function.* Simon Fraser University : Virtual Library of Simulation Experiments, 2013.

[6] Bingham, D. Surjanovic, S. *Test Functions and Datasets: Rastrigin Function.* Simon Fraser University : Virtual Library of Simulation Experiments, 2013.

[7] Bingham, D. Surjanovic, S. *Test Functions and Datasets: Michalewicz Function.* Simon Fraser University : Virtual Library of Simulation Experiments, 2013.

[8] Croitoru, E. *Teaching: Genetic Algorithms.* Romania [Iasi] : Computer Science UAIC.

[9] R.N. Greenwell, J.E. Angus, M. Finck *Optimal mutation probability for genetic algorithms.* sciencedirect.com, 1995.

[10] S. Sarmady *An Investigation on Genetic Algorithm Parameters.* sarmady.com, 2023.