Improved Genetic Algorithm for solving Single Objective Bound Constrained Numerical Optimization Problems

George Stoica
Faculty of Computer Science
”Alexandru Ioan Cuza” University
Iași, Romania
george.stoica@info.uaic.ro

***Abstract*—Single objective bound constrained problems represent a well studied field due to numerous real-world applications and are still a challenging task. Different Evolutionary Algorithms (EAs) have been proposed to solve such problems, however the standard Genetic Algorithm (GA) is not between the top competitors. While carefully fine-tuned GAs are able to solve specific optimization problems, no GA is known for being able to solve a wide range of them. This paper aims to explore improving the standard GA in order to solve the CEC 2022 Single Objective Bound Constrained Numerical Optimization problems through hybridization, encoding change and multi-operators. The improvements make the GA perform better and even though it does not surpass other established solutions, they open the possibility to further research and refine it to solve these kind of problems.**

***Index Terms*—optimization, nature inspired methods, genetic algorithms, CEC 2022**

I. Introduction

Many optimization problems can be found in real-life situations and some of them can be solved by either minimizing or maximizing one or more objective functions. Deterministic algorithms are usually not able to solve such problems, due to their computational complexity. Population based methods however are able to give good results due to their stochastic approach and by mimicking an optimization process inspired from nature.

Genetic Algorithms (GAs) are one of the often ignored candidates for solving these problems, mainly because other nature inspired methods such as evolutionary algorithms (EAs) or differential evolution (DE) algorithms have proved to be more adequate.

In this paper the feasibility of GAs is addressed along with several modifications in order to improve their results. Applying a GA with the same parameters on a wide range of functions is known to give average results according to the No Free Lunch theorem [11]. Even so, it is possible to optimize an algorithm to solve specific classes of problems, such as the objective functions from the CEC 2022 competition. Therefore, enhancements of the GA are studied in order to make this method a better suited option for this situation.

The rest of the paper is organized as follows. In Section II the CEC 2022 competition and the benchmark functions are introduced, while in Section III previous submissions are analyzed. Section IV presents the fore-mentioned improvements to the GA and the motivation behind them. The experimental results and the setup is described in Section V and finally, the conclusion in Section VI.

II. Competition

The CEC 2022 competition consists of 12 minimization problems that are to be treated as black-box problems. The number of dimensions are either 10 or 20, and the functions to be minimized have their global optimum shifted and rotated. For 10 dimensions, the maximum number of function evaluations (maxFES) is 200’000, while for 20 dimensions maxFES is 1’000’000. The search range is between $[-100, 100]^D$.

Details of the competition and the functions used can be found in [6]. This paper uses only 8 of the 12 functions, the second and the third hybrid functions along with the third and the fourth composition functions have not been taken into consideration when running the experiments due to time constraints.

III. Previous solutions

Previous submission for this competitions have used variants of nature inspired algorithms such as EA [9], DE [8], [10], [4], [3], Cuckoo search [7] and hybrids like Multi-Population Exploration-only Exploitation-only Hybrid [2].

The lack of any solution which uses GA proves that the effort of optimizing this algorithm is higher and that the standard GA does not easily provide a set of parameters which give good results on all of these functions. Therefore a direction of research is to find ways to make improvements that lead to a better optimization process and this is the objective this paper aims to meet.

IV. Genetic Algorithm improvements

As it has been mentioned before, parameters for the GA can be adjusted in order to get good results for a specific problem. But a good set of parameters that minimize all functions together would not have adequate results for each function in particular. In order to increase the optimization power of the standard GA several strategies have been implemented.

Due to the fact that the populations in GA is represented by chromosomes made up from a sequence of genes, the encoding is a significant factor which can affect finding good solutions. Moreover, both mutation and crossover behave differently for different encodings, provided that the Hamming wall is different.

Several encoding types and their effect has been studied in [1], in which the performance of quaternary and octal encodings along with binary and value encodings is analyzed. The authors have argued that running the GA with different encodings can lead to finding better solutions. In this paper only the binary encoding and the Gray encoding have been considered.

This research proposes to alternate the two encodings during the algorithm. By doing this, the exploration is enhanced at each change because the same mutation could find better values in the same direction that was previously ignored or not represented enough in the population. Since the space of the representation has been changed, previously inaccessible places can now be explored. As a consequence, local optima are easier to escape from.

While it might have been a better idea to use a multi-population approach, each population with it's own encoding, in this paper there is only one population and the encoding is changed for all the chromosomes using a scheduler. Furthermore, another improvement which intends to enhance the exploratory part, adaptive mutation, has been used to increase the mutation rate for one generation, also using a scheduler.

Improvements to the exploratory part have also been done. In this sense, two strategies of selecting chromosomes for crossover have been tested. However, the type of crossover used was single point crossover and other possibilities were not tested. Besides the classic strategy, the second proposed method is to let each chromosome do crossover with any other chromosomes, including those that have been through crossover before in the current generation. However, experimental results have not been able to prove one method superior to the other. Another point of study would have been to change the crossover operator and use the xor function instead of swapping genes, but this has not been yet studied due to time constraints.

Nevertheless, hybridization with the hillclimbing algorithm has been used. However, hillclimbing has been applied only on the best chromosome at the end of the algorithm due to the fact that it would consume a big part of the function estimations allowed in the competition if used more times. Even so, hillclimbing is used together with the encoding change such that the best chromosome reaches it's local minima in both representational spaces. Changing the encoding in the hillclimbing algorithm has given better results than applying it only on one encoding.

The selection of the population is done using elitism combined with the wheel of fortune selection, using the selection pressure as the parameter.

V. Experimental results and analysis

The programming language used to implement the GA and run the experiments is C++, using the C++20 standard and compiled using gcc 11.2.0.

The experimental results are divided into 3 groups. The first one consists of the results obtained using the classic GA, without any improvements. The second one is the group that uses the encoding change, while the third one is the group that use hillclimbing on top of it. The third group does not respect the conditions of the competition, because all experiments are obtained by exceeding the maxFES limit.

TABLE I
Classic GA – 10 dimensions

Function	***Mean***	***Median***	***Std***	***Min***	***Max***
levy	3.93	3.76	0.95	2.25	6.67
schaffer_F7	5.38	5.22	1.50	2.89	9.06
rastrigin	25.39	21.08	12.67	6.21	51.97
rosenbrock	22.97	21.03	8.51	14.69	58.42
zakharov	5748.71	5136.53	2317.79	2822.96	13733.8
hf01	30128.18	28780.6	20668.96	2007.08	87005.2
cf01	1130.98	1130.99	0.05	1130.88	1131.1
cf02	37724.28	37701	47.53	37689.7	37831.6

All three experiments are run on 8 functions for 10 and 20 dimensions. The same parameters are used in all situation, except for the specific ones introduced by mutation change and hillclimbing. The parameters were chosen by running the classic GA on 20 dimensions and minimizing the sum of all function results. Therefore, the result for the first 4 functions would have a small corresponding weight, and as it can be deduced from the experiments *hf01* was the main influencer and gainer from the optimization process.

TABLE II
Classic GA – 20 dimensions

Function	***Mean***	***Median***	***Std***	***Min***	***Max***
levy	29.78	30.33	5.06	19.36	40.89
schaffer_F7	27.60	27.58	4.69	18.09	36.05
rastrigin	62.66	58.46	23.47	22.98	137.94
rosenbrock	73.75	73.87	3.11	66.22	78.83
zakharov	20117.5	20353.3	3712.3	12359.4	26055.2
hf01	3063562	3150685	1075713	1616340	6559920
cf01	7635.08	7635.08	0.73	7632.9	7636.57
cf02	159219	159179	77.78	159166.0	159413

The parameters used are 0.9 probability for crossover, 0.005 probability for mutation, with 0.025 being the increased mutation probability which happens once 20 epochs. The population size is 100, and 2 of the chromosomes are elites, while the other are selected from using the wheel of fortune rule with a selection pressure of 10 when calculating the fitness for chromosome i as shown in (1).

$$f_i = \left(\frac{max - f_i}{max - min + \epsilon} + 1 \right)^{selection\_pressure} \tag{1}$$

The first group of experiments (Fig. I, Fig. II) show that the two functions formed through composition seem to have the

TABLE III
GA WITH ENCODING — 10 DIMENSIONS

Function	*Mean*	*Median*	*Std*	*Min*	*Max*
levy	4.01	3.93	1.13	1.51	6.11
schaffer_F7	6.48	5.40	6.78	2.68	41.34
rastrigin	25.82	24.63	9.69	10.27	57.92
rosenbrock	16.62	16.62	4.29	9.55	31.51
zakharov	5278.45	4261.24	2864.28	1981.08	14917.7
hf01	26559.24	19366.4	22145.9	3187.78	86834
cf01	1130.88	1130.87	0.04	1130.80	1130.96
cf02	37692.41	37691.75	2.25	37689.60	37698.2

entire population consistently stop inside the same plateau, and this phenomena can be also be observed in the latter experiments. Therefore most of the effort was directed towards optimizing the hybrid function and zakharov's function to the detriment of the first four methods.

TABLE IV
GA WITH ENCODING — 20 DIMENSIONS

Function	*Mean*	*Median*	*Std*	*Min*	*Max*
levy	34.45	34.74	5.73	21.83	49.31
schaffer_F7	21.53	21.63	4.14	12.48	30.28
rastrigin	70.05	71.84	15.19	37.43	96.11
rosenbrock	68.01	68.57	3.03	61.56	73.03
zakharov	20634.3	20334.3	3560.27	14048.8	28561
hf01	3393617	3202775	1232067	1452480	6580690
cf01	7633.87	7633.81	0.32	7633.36	7634.54
cf02	159176	159176	4.8	159165	159187

The second group of experiments has the same parameters with the addition of the encoding change each 7 epochs. As Fig. I and Fig. II show, the mean value for the first four functions has decreased. This is because they would have preferred an encoding change rate between 25 and 50 epochs. However, zakharov's function and the hybrid function which had bigger weights have modified the parameter towards a lower value. Encoding change also has a small effect on the composition functions, bringing their mean to a slightly lower value while lowering their standard deviation. It can seen that both functions reached a better local optima, even though they have not escaped the plateau.

TABLE V
GA WITH ENCODING AND HC — 10 DIMENSIONS

Function	*Mean*	*Std*	*Min*	*Max*	*FES*
levy	1.42	0.90	0.00	3.43	264382
schaffer_F7	2.51	1.35	0.43	4.90	2080049
rastrigin	24.38	9.99	8.95	45.77	206815
rosenbrock	3.87	3.76	0.00	8.92	4826870
zakharov	0.00	0.00	0.00	0.00	371865
hf01	145.14	321.35	4.31	1310.1	577881951
cf01	1130.65	0.00	1130.65	1130.65	209047
cf02	37689.34	2.02	37687.2	37695.4	206024

In the third group of experiments, a column with the function evaluations (FES) has been added to illustrate how many hillclimbing steps have been applied to the best chromosome after executing maxFES steps with the GA. The hillclimbing gives a better result due to the fact that it searches on both accepted representations. A huge improvement is obtained for the zakharov function, which manages to reach a value close to 0, even when rotated and shifted. Compared to the previous results, it's clear that the mutation probability was too high for the zakharov function to be minimized, which suggests that a strategy of minimizing the mutation across the algorithm might give good results for this method. The hybrid function also experienced a huge improvement, in exchange for a huge number of function evaluations. Rastrigin's function however is not minimized by the hillclimbing algorithm, and neither the two composition functions.

TABLE VI
GA WITH ENCODING AND HC — 20 DIMENSIONS

Function	*Mean*	*Std*	*Min*	*Max*	*FES*
levy	11.08	3.02	5.97	18.27	1343748.43
schaffer_F7	12.15	3.79	5.05	18.58	30216334.7
rastrigin	60.56	18.59	27.86	103.7	1017881.7
rosenbrock	47.97	1.88	44.90	49.08	1139761.53
zakharov	0.00	0.00	0.00	0.00	2550144.67
hf01					
cf01	7628.84	0.00	7628.84	7628.84	1021324.5
cf02	159149.57	2.64	159146	159157	1023119.8

All experiments have 30 runs, except the hybrid function for 20 dimensions, which has none due to time constraints.

VI. CONCLUSION AND FUTURE WORK

While the results obtained are not good and the preferred solutions to this competition use DE or EA, this paper explores ways of improving the results obtained using the GA. Changing the encoding is one method that can be applied, moreover using a more intelligent approach such as using a multi-population GA instead of changing the encoding for all chromosomes at once. Moreover, trying quaternary, octal and even value encoding in different populations might solve and boost the exploratory power and direction of the GA. Furthermore, considering that the easier function's results were affected by the more complex ones, a smarter way of adjusting the parameters could be considered in order to minimize them better.

This paper also tried to use different strategies for the selection of chromosomes during crossover, which didn't show any improvement, but might do so in a multi-population environment. Nevertheless, there is room for further research in optimizing the standard GA in order to make it a suitable target for solving more problems. Future research could include, besides the multi-population with multiple encodings, different operators for each population that would lead to an enhanced exploration and exploitation.

REFERENCES

[1] Serkan Bekiroğlu, Tayfun Dede, and Yusuf Ayvaz. Implementation of different encoding types on structural optimization based on adaptive genetic algorithm. *Finite Elements in Analysis and Design*, 45(11):826–835, 2009.

[2] Antonio Bolufé-Röhler and Stephen Chen. A multi-population exploration-only exploitation-only hybrid on cec-2020 single objective bound constrained problems. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8. IEEE, 2020.

[3] Janez Brest, Mirjam Sepesy Maučec, and Borko Bošković. Single objective real-parameter optimization: Algorithm jso. In *2017 IEEE congress on evolutionary computation (CEC)*, pages 1311–1318. IEEE, 2017.

[4] Petr Bujok and Josef Tvrdík. Enhanced individual-dependent differential evolution with population size adaptation. In *2017 IEEE congress on evolutionary computation (CEC)*, pages 1358–1365. IEEE, 2017.

[5] Eugen Croitoru. Details about the ga and it's implementation, available at https://profs.info.uaic.ro/∼eugennc/teaching/ga/.

[6] Abhishek Kumar, Kenneth V. Price, Ali Wagdy Mohamed, Anas A. Hadi, and P. N. Suganthan. Problem definitions and evaluation criteria for the cec 2022 special session and competition on single objective bound constrained numerical optimization. In *https://github.com/P-N-Suganthan/2022-SO-BO/blob/main/CEC2022%20TR.pdf*.

[7] Rohit Salgotra, Urvinder Singh, Sriparna Saha, and Amir H Gandomi. Improving cuckoo search: incorporating changes for cec 2017 and cec 2020 benchmark problems. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–7. IEEE, 2020.

[8] Karam M Sallam, Saber M Elsayed, Ripon K Chakrabortty, and Michael J Ryan. Improved multi-operator differential evolution algorithm for solving unconstrained problems. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8. IEEE, 2020.

[9] Karam M Sallam, Saber M Elsayed, Ruhul A Sarker, and Daryl L Essam. Multi-method based orthogonal experimental design algorithm for solving cec2017 competition problems. In *2017 IEEE Congress on Evolutionary Computation (CEC)*, pages 1350–1357. IEEE, 2017.

[10] Vladimir Stanovov, Shakhnaz Akhmedova, and Eugene Semenkin. Ranked archive differential evolution with selective pressure for cec 2020 numerical optimization. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–7. IEEE, 2020.

[11] David H Wolpert and William G Macready. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1):67–82, 1997.