A New Approach Method of Crossover Process Based On Genetic Algorithm Using High Dimensional Benchmark Functions

Mustafa TUNAY
Department of Computer Engineering
Istanbul Gelisim University
Istanbul, Turkey
mtunay@gelsim.edu.tr

***Abstract*— The design of the improved genetic algorithm (GA+) is based on a meta-heuristic search for optimization problems. In this paper, the crossover process in the original genetic algorithm is improved. The improvement of the crossover process is renewed by applying two conditions. One of them is keeping the last genes (constant) for each population; the second one is about rotating genes according to the defined range of points between each two selected populations. The improved genetic algorithm (GA+) has the possibility of accelerating local convergence. Therefore, it gets a chance to search for better values globally using these conditions. All processes in the improved genetic algorithm have been represented in this paper. The performance of the proposed algorithm is evaluated using 7 benchmark functions (test functions) on different dimensions. Ackley function, Rastrigin function and Holzman function are multi-modal minimization functions; Schwefel 2.22 function, Sphere function, Sum Squares function and Rosenbrock function are uni-modal minimization functions. These functions are evaluated by considering cases that are minimized by having a set of dimensions as 30, 60, and 90. Additionally, the performance of the GA+ is compared with the performance of comparative optimization algorithms (meta-heuristics). The comparative results have shown the performance of the GA+ that performs much better than others for optimization functions.**

Keywords—Benchmark Functions, Genetic Algorithms, Improved Crossover Process, Metaheuristic Search.

I. Introduction

In this section, a literature review of optimization algorithms [1-5] is given. The research-based on evolutionary algorithms [6-8] including differential evaluation [9,10] and especially genetic algorithms [11-13] have been analyzed. The design of the high accuracy genetic algorithm for solving dimensional optimization problems is noticed.

Optimization issues usually need to use mathematical algorithms for seeking out a good solution iteratively in analytical solutions. In this spirit, different optimization strategies have been designed for finding a good solution. There is an optimization method which is simultaneous perturbation stochastic approximation (SPSA) for multivariate optimization. This optimization technique finds a good solution in issues such as feedback control, simulation optimization, image processing, adaptive modeling, estimation of distribution algorithms, atmospheric modeling. The proposed method uses gradient approximation in any case of the dimension of the optimization problem. The SPSA method decreases especially in problems that need to be optimized due to the cost of optimization solutions. More details are referred to in [14].

Many optimization problems [15-17] are primarily to find the best solution within their specific ranges. This kind of optimization problem usually refers to the best solution functions for solving using applied mathematics functions. Optimization problems include searching "the best solution" from the values of some objective function ranges for different types of objective optimization. The solutions of nonlinear optimization acquire great importance.

The mathematical model [18,19] is applied to many fields as economics, industry, computer science, game theory, artificial intelligence (AI), and many other areas in the real world. Many mathematicians studied to explore a wide range of complex tasks and focused on systems with multiple factors that they interact in nonlinearly. They obtained two cases as a result of this study. They are major effects of co-adaptation and co-evolution. Thus, the mathematical model describes how to change the traditional process of mathematical genetics.

The traditional process of optimization techniques is applied for obtaining the solution of different optimization problems. Traditional computational intelligent systems are based on private "internal" cognitive and computational processes. However, the traditional optimization techniques are based on finding the derivative of objective functions that means a locally optimized solution. Additionally, there are many other various problems in the proposed techniques that do not fare well for finding the global optimal solution of the functions. Many multi-objective applications of evolutionary algorithms have found increasing applications in the domain of data mining problems.

A new heuristic approach is applied to minimize nonlinear and non-differentiable continuous space functions [20]. The proposed algorithm selects the difference between two vectors randomly. Additionally, the proposed algorithm perturbs an existing vector in chosen population vectors. The perturbation is done for every population vector. The proposed method is demonstrated to converge faster for multi-objective optimization.

Genetic algorithms provide complex adaptive systems for economic theory using machine learning methods. Adaptation is a biological process that is to survive in environments confronting organisms that evolve by rearranging genetic material. Several scientists studied to seek out a solution for nonlinearity. Holland presents a mathematical model for complex interactions [21].

One of the widely used adaptive heuristic search algorithm used for multi-objective optimization. The proposed algorithm relies on the evolutionary conception of natural selection [22]. Natural evolution is randomly generated by individuals from a population.

There is also a viable new approach to stochastic combinatorial optimization which is inspired by the behavior of the ant. The proposed algorithm's main features are constructive greedy heuristic, distribution of computation, and positive feedback. Firstly, the greedy heuristic finds acceptable solutions for the search process. Secondly, the distribution of computation avoids premature convergence. Finally, the positive feedback explains the fast discovery of the best solutions. The proposed methodology is applied in practical problems to solve a set of problems for the robustness of the approach [23].

The genetic algorithm builds using the differing qualities of a new strategy in the detection of epileptic seizures from electroencephalogram (EEG). The detection of epileptic seizures from electroencephalogram transforms packets including energy, entropy, kurtosis, and skewness using discrete wavelet for creating features of signals. Clinical EEG data is commonly used in the experiment of epileptic and normal subjects. The proposed is improved GA with a support vector machine (GA-SVM). This means it is a new method for finding a good solution using a hybrid approach with wavelet packet decomposition [24].

The design of the tactical berth allocation problem (TBAP) is a biased random key in the modification of the genetic algorithm. It contains both the minimization of the housekeeping expenses; the first one is from the transshipment compartment streams in the middle of ships, the second one is about the amplification of the aggregate estimation of the quay crane profiles doled out of the ships. The acquired results for handling the TBAP have demonstrated that the proposed calculation is appropriate to proficiently take care of this issue [25].

A new structured population approach, which is a hierarchy of hypercube is represented as the population of GA. This approach generally leads to a more superior performance than palmitic GA [26]. Additionally, this research does not build sub-populations that are based on the information of the genes of individuals. The structure of subpopulations could help to achieve better performance and a more efficient searching strategy. The proposed approach can build the structure of a population by dividing the searching space.

The improved artificial chromosomes with a genetic algorithm (ACGA) is a new tendency for search optimization. The proposed algorithm has been applied to real-world problems successfully for solving scheduling problems. However, ACGA can not perform well in some scheduling problems. It does not consider variable interactions if sequence-dependent setup times are considered particularly. Thus, the previous one will improve variable interactions to influence the processing time. The improvement of ACGA is successfully applied to single machine scheduling problems. This improvement of ACGA is improved with a bi-variate probabilistic model. This is called the design of ACGA II. It includes some heuristics and local search algorithms and variable neighborhood search (VNS) [27]. The proposed method is successfully demonstrated to solve single machine scheduling problems with sequence-dependent setup times for the dating environment.

Many real-world optimization problems are using mathematical algorithms. It seeks an iterative solution because the function or the constraints of the objective problem can be improved over time. If these cases are undefined past in the optimization process, we are called dynamic for these types of problems. There are some difficulties in optimizing dynamic environments with the goal that the calculations for rationalization in these situations would be to use some systems keeping in mind the ultimate objective of overcoming difficulties. There are many algorithms for optimization problems.

There is a new technique for crossover operator. This is called an inversed bi-segmented Average Crossover (IBAX). It improves the offspring generation of the genetic algorithm for variable and numerical optimization problems. It attempts to come into view with a new mating scheme that in generating new offspring is under the crossover function using the IBAX operator. It has a more efficient and optimization solution for variable minimization on premature convergence problem using GA [28, 29].

There is another optimization problem for the bin packing problem (BPP). GA is one way to solve the Bin Packing (BPP) problem. The goal of BPP is to minimize the number of containers used by maximizing their content. A combination of BPP and GA is applied to the printed paper in digital printing. GA improvement is enhanced by random crossover and dynamic mutation. With this application, GA performance in the case of BPP can solve the problem of premature convergence and maximize print distribution [30].

The genetic algorithm is improved by a new technique for planning collision-free paths with static obstacles. The improved genetic algorithm is realized in the crossover process. This process has an important rule about the fitness value of the progeny. It should compare with the fitness value of the parents. Thus, the parents (chromosome) of the best fitness value will be needed for the mutation process. Meanwhile, the worse fitness value will be excluded from the best fitness value completely. This technique is applied to the intelligent navigation system for autonomous mobile robots such as differential drive mobile robots [31].

As it was mentioned above different multi-objective evolutionary algorithms such as genetic algorithms (GAs) and differential evaluation algorithms have been designed. These algorithms have found many practical applications. These algorithms have been applied in optimization issues successfully to solve many difficult optimization search problems. Many improvements have been done to develop optimization to search for the best solution to the problems.

In this paper, the improved genetic algorithm includes the selection process, the improved crossover process, and the mutation process. The crossover process in the original genetic algorithm is improved with a new technique that is keeping the last genes for each population and applying processing as rotating genes according to the assigned random numbers. These numbers define the range of genes for all populations. After that, they are replaced one by one between the two selected populations. It will continue between the selected populations (groups of two) with the same specific locations in them. Thus, the improved genetic algorithm (GA+) has the possibility of accelerating for local convergence and it gets a chance to search for better values globally. For all processes in the improved genetic algorithm have been represented in this paper. Moreover, the performance of the GA+ was tested on some benchmark functions. The performance of the GA+ has shown much better convergence rates than comparative optimization algorithms.

The organization of this article is included as follows: Section 2 describes the improved genetic algorithm (GA+). Section 3 describes the test functions. The information of seven benchmark functions are given. The performance of proposed algorithm is tested on seven test functions and is compared with metaheuristic algorithms on different dimensional functions. Section 4 describes the conclusion in this article.

II. Related Work

In this study is mainly presented the improvement of the crossover process that is renewed by applying two conditions. One of them is keeping the last genes (constant) for each population. The second one is rotating genes according to the defined range of points (the assigned two random numbers). The details regarding the pseudo-code of the GA+ are illustrated in Algorithm 1.

```
Start
    Initialization population (Ps)
        Done= false
While not done Do
Calculate the fitness of each individual population
PS = PS+1
    Selection (PS -1)
        Modification Crossover Cp
        *Keeping last gene constant for each population
        *Rotating genes according to the defined range of points
        *Rapleced them with other one
        Mutation Mp
    Done = Optimization
Stop?
End While
Display `all best solution`
End
```

Algorithm 1. The pseudo-code of the improved genetic algorithm

The GA+ is based on the meta-heuristic search for optimization problems. The main steps of the GA+ consist of selection process, improved crossover process, and mutation process. All processes in the improved genetic algorithm have been designed in Figure 1.

The improvement of the crossover process is explained in all details in the next section.

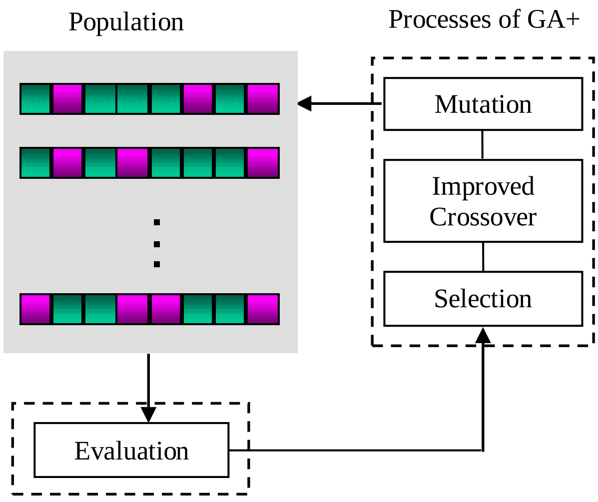

Fig. 1. The flow-chart of the GA+

2.1 *Selection Process*

The aim of the selection process selects good solutions and eliminates bad solutions in a population. This process is a genetic operator used in the improved genetic algorithms to select potentially useful solutions for recombination. This process uses the "fitness function" for assigning fitness to possible solutions or chromosomes. It is in fitness proportionate selection, as in all selection methods.

The fitness function is used to associate a probability of selection with each chromosome. If it is the fitness of individual i in the population, its probability of being selected is:

$p_i = \frac{f_i}{\sum_{j=1}^{N} f_j}$ where N is the number of individuals in the population. (1)

2.2 *Improved Crossover Process*

The crossover process is a genetic operator that combines two individuals. It produces a new population according to the specific processing rules. There are several kinds of crossover operators such as two-point crossover, multi-point crossover, and uniform crossover.

Firstly, the GA+ begins with the generation of the population and all populations are generated randomly. All these populations are matched (the selected groups of two; population 1 and population 2) to each other randomly. The two random numbers are generated randomly between the length of the population and are determined by the range of location of columns for each population. That is, the assigned two random numbers are defined for the range of genes. After all, these genes in the specified range of location of columns are rotated, then these rotated genes are replaced between the selected two populations one by one with a defined range of location of the columns that are formed in the same way. After that, all the last constant genes in the population will be replaced between the selected ones one by one according to the same location of columns. It will keep going between the selected populations (groups of two) with the same specific locations in them. The detail regarding the visualization of the new approach of the improvement of the crossover process is shown in Figure 2.

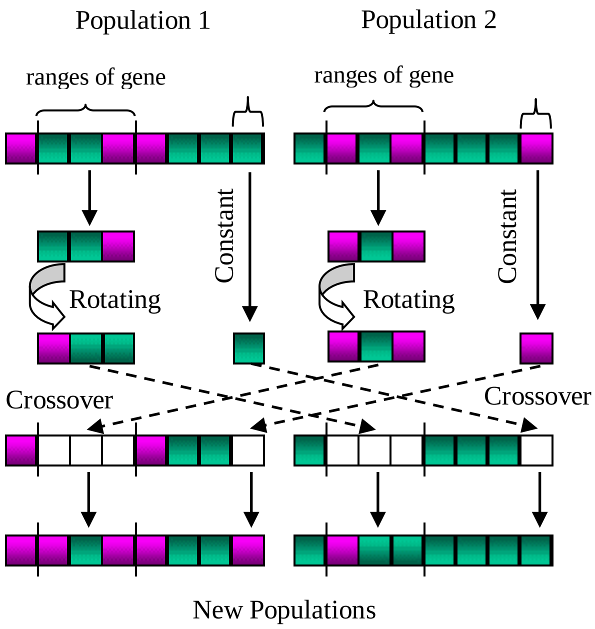

Fig. 2. The new approach of the improved crossover process for two new populations

Secondly, there is an important point about defining a value of the specific number in the modification of the crossover process for creating new chromosomes in a population. If the length of the population is greater than the value of the defined specific number, the population is divided into new pieces of chromosomes according to the value of the defined specific number and for each chromosome in the population have the same size (length for each chromosome). Provided that the last genes for each new chromosome in the population should be kept constant. The two random numbers are generated randomly between the length of chromosomes in the population and are determined the range of location of columns for each chromosome in the population. That is, the assigned two random numbers are defined for the range of genes for each chromosome in the population. After all, these genes in the specified range of location of columns are rotated, these rotated genes are replaced between the selected two populations (for each chromosome in population) one by one with a defined range of location of the columns. After that, the last genes for each chromosome in the population will be replaced between the selected ones one by one according to the same location of columns (the range of genes in the chromosome). It will keep going between the selected populations (groups of two) with the same specific locations in them.

The detail regarding the visualization of the new approach of the improvement of the crossover process for a piece of chromosomes in a population is shown in Figure 3.

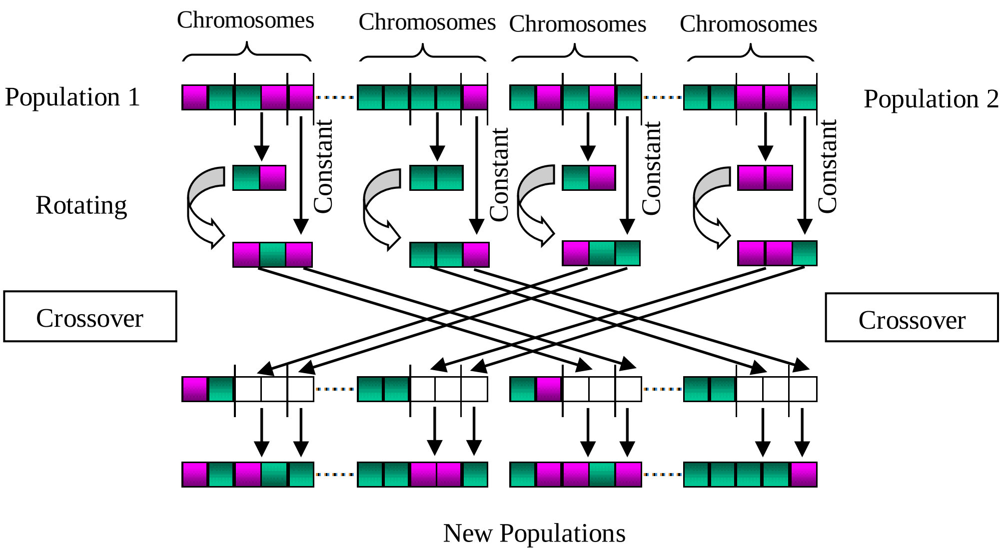

Fig. 3. The new approach of improved crossover process for a piece of chromosomes in a population

2.3 *Mutation Process*

This process occurs at each position in a bit string with a specific probability. This specific probability is generally defined between 0.1 or less according to the length of the population. The principle of this process assigned anyone a random number. According to the assigned value of the number is defined which location of the column in population. If its value is "1", it will be "0"; otherwise. The details regarding the visualization of the mutation process are shown in Figure 4.

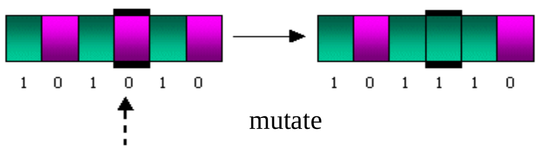

Fig. 4. Mutation process

However, there is an important point for defining the length of the population. If the length of the population is greater than the value of the defined specific number, the population is divided into new pieces of chromosomes according to the value of the defined specific number and each chromosome in the population has the same size (length for each chromosome). The principle of this process is to assign a random number in the mutation process. According to the assigned value, the number is defined for each chromosome in the population, that is, it will be the locations of the column in chromosomes for each population. If its value is "1", it will be "0"; otherwise.

III. Test Functions

The performances of the improved genetic algorithm (GA+) was implemented on Matlab-Simulink Version 2017. The GA entails setting several parameter values. The primary parameters of genetic algorithms included as crossover rate (0.60) and mutation rate (0.1). The features of the hardware and software tools of the computer used are as follows; CPU: Intel (R) Core (TM) i3-4005U M, SPEED: 1.70 GHz – x64, RAM: 4.00 GB and OS: Microsoft Windows 8.1. The improved genetic algorithm (GA+) performed on specific benchmark functions (test functions).

They consists of seven optimization test functions, namely, Ackley function (F1), Rastrigin function (F2), Schewefel 2.22 function (F3), Sphere function (F4), Sum Squares function (F5), Holzman function (F6) and Rosenbrock function (F7). For more information about these benchmark functions including implementation codes and more, we refer to the reader; http://benchmarkfcns.xyz/fcns.

Ackley function, Rastrigin function and Holzman function are multi-modal minimization functions; Schwefel 2.22 function, Sphere function, Sum Squares function and Rosenbrock function are uni-modal minimization functions. The properties of these functions are briefly shown in Table 1 as equation of test functions and range. The performances of the GA+ is evaluated with the use of these functions in the next section.

3.1 The Performance of GA+ on Test Functions

The performance of proposed algorithm was compared on seven test functions in this section. The GA+ was evaluated to minimize functions having the set of dimensions as 30, 60, and 90. The optimization experiments of the proposed algorithm (GA+) was performed on the three different dimensions for the 7 benchmark optimization functions. The parameters for the algorithm was the set as follows: size of population = 50, iterations = $5x10^2$ and 10^3 and dimension (*D*)= 30, 60 and 90.

TABLE 1
The GA+ performed on test functions for dimensions as 30, 60 and 90

Test Functions		**30 Dimension**		**60 Dimension**		**90 Dimension**	
		GA+		GA+		GA+	
		500 iterations	1000 iterations	500 iterations	1000 iterations	500 iterations	1000 iterations
F1	*Best*	1.28e-13	1.28e-13	1.14e-13	1.14e-13	1.14e-13	1.17e-13
	Mean	6.45e-06	5.59e-10	1.70e-12	2.23e-12	2.05e-08	1.05e-11
	Std.	2.75e-05	2.05e-09	6.50e-12	9.25e-12	8.88e-08	1.40e-11
F2	*Best*	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
	Mean	1.60e-04	4.75e-06	8.88e-16	0.00e+00	3.65e-15	0.00e+00
	Std.	5.80e-04	1.75e-05	3.89e-15	0.00e+00	1.59e-14	0.00e+00
F3	*Best*	4.44e-13	1.25e-12	4.73e-27	4.73e-27	7.09e-27	7.09e-027
	Mean	2.92e-05	2.45e-05	8.29e-23	5.15e-27	6.19e-17	4.67e-017
	Std.	8.60e-05	8.06e-05	1.79e-22	0.00e+00	1.39e-16	2.18e-016
F4	*Best*	7.49e-28	7.49e-28	1.34e-27	1.18e-27	1.94e-27	1.77e-27
	Mean	7.70e-06	1.97e-07	3.13e-22	1.35e-24	1.51e-16	3.56e-18
	Std.	2.45e-05	8.08e-07	1.25e-21	0.00e+00	6.44e-16	1.49e-17
F5	*Best*	2.18e-21	3.44e-26	1.39e-25	1.39e-25	3.32e-25	3.15e-25
	Mean	2.89e-04	2.15e-05	8.40e-22	2.65e-22	1.59e-15	2.32e-18
	Std.	8.33e-04	8.19e-05	3.66e-21	6.90e-22	4.30e-15	1.50e-17
F6	*Best*	7.19e-54	1.31e-53	1.10e-53	1.10e-53	2.49e-53	2.49e-53
	Mean	1.59e-08	7.29e-16	2.88e-37	3.83e-44	3.47e-25	5.15e-31
	Std.	6.95e-08	2.40e-15	0.00e+00	0.00e+00	0.00e+00	0.00e+00
F7	*Best*	2.82e+01	2.82e+01	5.80e+01	5.76e+01	8.83e+01	2.82e+01
	Mean	2.88e+01	2.88e+01	5.86e+01	5.86e+01	8.87e+01	8.86e+01
	Std.	1.92e-01	1.45e-01	2.60e-01	3.59e-01	2.92e-01	2.33e-01

The performance of the improved genetic algorithm was evaluated the same dimensions (30, 60, 90) and as well using a varying number of iterations in solving seven benchmark functions. The properties of a benchmark functions having standard parameters were implemented on Matlab program. These functions are summarized as the best, the mean, the standard deviation and evaluated over successful 100 runs. The performance of the improved proposed algorithm is shown in Table 1.

The algorithm, which finds the best solution and solves optimization problems is designed. The design of the improved crossover process in a convergence state help to the best position to jump out of possible local optimal solution to further increase the performance of proposed algorithm. Thus, the search strategy in the proposed algorithm has proven to be a success global optimal solution, convergence optimal solution, allows speeding the learning of the system with faster convergence rates for all these optimization problems.

The performance of the GA+ is also compared with the performance of meta-heuristic algorithms in the next section.

3.2 Comparison between GA+ and Metaheuristic Algorithms

This section presents the comparison of the performance of the proposed algorithm with meta-heuristics algorithms. Examples of meta-heuristic algorithms include the ant colony optimization (ACO), the bat algorithm (BAT), particle swarm optimization (PSO), ant lion optimizer (ALO), krill herd (KH), monarch butterfly optimization (MBO) and moth-flame optimization (MFO). Other metaheuristics have also been developed based on the evolutionary theory including differential evolution (DE). The above meta-heuristics are classified as stochastic optimization techniques. All algorithms were evaluated by considering the cases in which functions having the set of dimensions as 30, 60, 90 for 50 iterations and averaged over 100 experimental runs. The population size is also set to 50.

TABLE 2
THE GA+ COMPARED WITH METAHEURISTIC ALGORITHMS

Test Functions	D	ACO	BAT	DE	PSO	GA+
F1	*30*	1.85E+001	1.99E+001	1.87E+001	1.87E+001	**7.32e-002**
	60	1.90E+001	1.99E+001	1.90E+001	1.90E+001	**2.87e-005**
	90	1.91E+001	1.99E+001	1.91E+001	1.91E+001	**1.98e-004**
F2	*30*	1.63e+002	4.34e+002	1.73e+002	1.73e+002	**2.89e+000**
	60	3.74e+002	9.33e+002	3.99e+002	4.00e+002	**4.90e-008**
	90	6.03e+002	1.44e+003	6.41e+002	6.31e+002	**7.67e-005**
F3	*30*	1.13E+002	2.95E+012	5.38E+001	1.14E+002	**2.39e-001**
	60	2.48E+002	2.29E+028	1.71E+002	2.49E+002	**3.00e-007**
	90	3.88E+002	6.75E+043	2.97E+002	3.89E+002	**5.91e-004**
F4	*30*	1.63E+002	1.67E+002	2.79E+001	5.12E+001	**9.21e-003**
	60	3.76E+002	3.91E+002	1.74E+002	2.13E+002	**6.14e-006**
	90	6.02E+002	6.19E+002	3.80E+002	4.29E+002	**5.65e-004**
F5	*30*	9.37E+003	9.27E+003	1.29E+003	2.30E+003	**1.79e-001**
	60	4.39E+004	4.29E+004	1.65E+004	1.62E+004	**1.29e-007**
	90	1.03E+005	1.03E+005	5.51E+004	5.06E+004	**3.70e-003**
F6	*30*	4.19E+005	4.19E+005	2.51E+004	6.73E+004	**1.80e-002**
	60	2.24E+006	2.13E+006	6.43E+005	9.31E+005	**6.05e-012**
	90	5.50E+006	5.24E+006	2.65E+006	3.19E+006	**1.79e-009**
F7	*30*	1.06E+008	2.32E+008	1.59E+007	2.25E+007	**3.04e+001**
	60	5.87E+008	5.97E+008	2.12E+008	2.11E+008	**5.89e+001**
	90	1.00E+009	9.99E+008	5.81E+008	7.39E+008	**8.89e+001**

The performances of all algorithms were compared using the same common parameters [32] and the same number of iterations for seven benchmark functions. The performance of the GA+ was compared with a selected collection of comparative algorithms that have been evaluated. The comparative algorithms are ACO, BAT, DE, and PSO. The comparative results demonstrated the performance of the GA+ which is also much better than the selected collection of the meta-heuristics algorithms (ACO, BAT, DE, and PSO) for seven standard benchmark functions. The proposed algorithm obtained 7.32e-02, 2.87e-05 and 1.98e-04 using Ackley function; 2.89e+00, 4.90e-08 and 7.67e-05 using Rastrigin function; 2.39e-01, 3.00e-07 and 5.91e-04 using Schwefel 2.22 function; 9.21e-03, 6.14e-06 and 5.65e-04 using Sphere function; 1.79e-01, 1.29e-07 and 3.70e-03 using Sum Squares function; 1.80e-02, 6.05e-12 and 1.79e-09 using Holzman function; 3.04e+01, 5.89e+01 and 8.89e+01 using Rosenbrock function having the set of dimensions as 30, 60 and 90 respectively. The best mean for each function is marked in bold and all details are shown in Table 2.

The performance of the GA+ was compared with other metaheuristic algorithms that have been evaluated. The comparative algorithms are ALO, KH, MBO, and MFO. The comparative results demonstrated the performance of the GA+ which is also much better than the selected collection of the other meta-heuristic algorithms for seven benchmark functions. The best mean for each function is marked in bold and all details are shown in Table 3.

The performance of the GA+ was compared with the performance of four comparative optimization algorithms, namely, GA, PSO, GAPSO, and the improved genetic particle swarm optimization algorithm (IGAPSO) using Rastrigin function and Sphere function having the set of dimension as 30. They are used here with the same parameters [33].

TABLE 3
THE GA+ COMPARED WITH OTHER METAHEURISTIC ALGORITHMS

Test Functions	D	ALO	KH	MBO	MFO	GA+
F1	*30*	1.37e+001	4.84e+000	1.41e+001	1.85e+001	**7.32e-002**
	60	1.53e+001	7.16e+000	1.60e+001	2.01e+001	**2.87e-005**
	90	1.62e+001	7.80e+000	1.57e+001	2.04e+001	**1.98e-004**
F2	*30*	1.29e+002	1.06e+001	9.96e+001	2.85e+002	**2.89e+000**
	60	3.79e+002	2.78e+001	2.37e+002	7.37e+002	**4.90e-008**
	90	6.64e+002	5.08e+001	4.47e+002	1.22e+003	**7.67e-005**
F3	*30*	1.06e+002	1.14e+001	5.24e+001	4.66e+002	**2.39e-001**
	60	1.31e+017	2.45e+014	1.44e+002	1.13e+017	**3.00e-007**
	90	1.57e+031	3.56e+027	2.74e+002	1.37e+032	**5.91e-004**
F4	*30*	1.57e+001	4.63e–001	6.38e+001	6.57e+001	**9.21e-003**
	60	5.10e+001	4.75e+000	1.93e+002	2.70e+001	**6.14e-006**
	90	8.84e+001	9.04e+000	3.58e+002	4.97e+002	**5.65e-004**
F5	*30*	7.56E+002	4.21E+001	5.24E+003	3.09E+003	**1.79e-001**
	60	5.44E+003	5.33E+002	2.51E+004	2.63E+004	**1.29e-007**
	90	1.39E+004	1.55E+003	6.47E+004	7.70E+004	**3.70e-003**
F6	*30*	3.23E+003	-2.21E+008	1.82E+005	8.48E+004	**1.80e-002**
	60	3.33E+004	-6.49E+012	1.08E+006	1.14E+006	**6.05e-012**
	90	1.06E+005	-1.65E+017	2.59E+006	3.76E+006	**1.79e-009**
F7	*30*	1.89E+006	8.69E+003	5.85E+007	4.74E+007	**3.04e+001**
	60	9.56E+006	2.28E+004	2.22E+008	3.61E+008	**5.89e+001**
	90	2.09E+007	3.85E+004	3.07E+008	7.55E+008	**8.89e+001**

TABLE 4
THE GA+ COMPARED WITH GA, PSO, GAPSO AND IGAPSO ALGORITHMS ON 30 DIMEMSION

30D	Iterations	Rastrigin Function Mean	Iterations	Sphere Function Mean
GA+	200	**1.83e-002**	200	**1.95e-005**
GA	401	8.596e+001	265	8.89e-002
PSO	373	1.200e+002	202	6.02e-003
GAPSO	361	9.05e+000	197	4.20e-004
IGAPSO	342	9.01e+000	186	4.12e-004

The comparative results suggest that the overall convergence rates of the GA+ still perform much better than others for Rastrigin function and Sphere function and obtained 1.83e-002 and 1.95e-005 respectively for $2x10^2$ iterations. Moreover, the best mean for each function is marked in bold and all details are shown in Table 4.

IV. CONCLUSIONS

The improved genetic algorithm (GA+) was designed for evaluation of the GA+ with the aim of the number of iterations. The number of iterations has shown obtained optimal solutions and convergence optimal solutions for optimization problems. The proposed algorithm recommended using a population of 50 as well as using a varying number of iterations ranging from 50, $5x10^2$ and 10^3 in solving specific benchmarking functions. The proposed algorithm obtained 7.32e-02, 2.87e-05 and 1.98e-04 using Ackley function; obtained 2.89e+00, 4.90e-08 and 7.67e-05 using Rastrigin function; obtained 2.39e-01, 3.00e-07 and 5.91e-04 using Schwefel 2.22 function; obtained 9.21e-03, 6.14e-06 and 5.65e-04 using Sphere function; obatined 1.79e-01, 1.29e-07 and 3.70e-03 using Sum Squares function; obtained 1.80e-02, 6.05e-12 and 1.79e-09 using Holzman function; obtained 3.04e+01, 5.89e+01 and 8.89e+01 using Rosenbrock function having dimensions as 30, 60 and 90 respectively for 50 iterations. Secondly, the proposed algorithm obtained 6.45e-06, 1.70e-12 and 2.05e-08 using Ackley function; obtained 1.60e-04, 8.88e-16 and 3.65e-15 using Rastrigin function; obtained 2.92e-05, 8.29e-23 and 6.19e-17 using Schwefel 2.22 function; obtained 7.70e-06, 3.13e-22 and 1.51e-16 using Sphere function; obtained 2.89e-04, 8.40e-22 and 1.59e-15 using Sum Squares function; obtained 1.59e-08, 2.88e-37 and 3.47e-25 using Holzman function; obtained 2.88e+01, 5.86e+01 and 8.87e+01 using Rosenbrock function for $5x10^2$ iterations. Finally, the proposed algorithm obtained 5.59e-10, 2.23e-12 and 1.05e-11 using Ackley function; obtained 4.75e-06, 0.00e+00 and 0.00e+00 using Rastrigin function; obtained 2.45e-05, 5.15e-27 and 4.67e-17 using Schwefel 2.22 function; obtained 1.97e-07, 1.35e-24 and 3.56e-18 using Sphere function; obtained 2.15e-05, 2.65e-22 and 2.32e-18 using Sum Squares function; obtained 7.29e-16, 3.83e-44 and 5.15e-31 using Holzman function; obtained 2.88e+01, 5.86e+01 and 8.86e+01 using Rosenbrock function for 10^3 iterations.

The improvement of the crossover process was renewed by applying two conditions. That is the method existing in the cross-over process renewed in the original GA. Briefly, the proposed algorithm keeps its original form without any external additions. For this reason, we believe that the GA+ is not very popular among researchers who compare with today's novel optimization algorithms in this study. However, the performance of GA+ was compared with many metaheuristic optimizers, including ACO, BAT, PSO, ALO, KH, MBO, MFO, DE, GA, GAPSO and IGAPSO; the set of comparative optimization algorithms as well as a collection of 11 algorithms. The comparative results included the best mean for the obtained optima. All those results showed an outstanding performance of GA+ in the majority of the evaluation cases in this paper.

REFERENCES

[1] R. Abiyev and M. Tunay, "Optimization of High Dimensional Functions through Hypercube Evaluation," *Computational Intelligence and Neuroscience*, volume 2015, pp. 1-11, 2015.

[2] M. Tunay, "Evolutionary Search Algorithm Based on Hypercube Optimization For High-Dimensional Functions," *International Journal of Computational and Experimental Science and Engineering (IJCESEN)*, 6(1), 42 – 62, 2020.

[3] R. Abiyev and M. Tunay, "Optimization Search Using Hypercubes," 2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Istanbul, Turkey, October 22-24, 2020.

[4] M. Tunay, "A New Design of Metaheuristic Search Called Improved Monkey Algorithm Based on Random Perturbation for Optimization Problems," *Scientific Programming*, volume 2021, pp. 1-14, 2021.

[5] R. Abiyev and M. Tunay, "Experimental Study of Specific Benchmarking Functions for Modified Monkey Algorithm," *Procedia Computer Science*, volume 102, pp. 595-602, 2016.

[6] J. Xiu , Q. He, Z. Yang and C. Liu, "Research on a multi-objective constrained optimization evolutionary algorithm," 2016 4th International Conference on Cloud Computing and Intelligence Systems (CCIS), Beijing, August 17-19, 2016.

[7] A. Slowik & H. Kwasnicka, "Evolutionary algorithms and their applications to engineering roblems," *Neural Comput & Application*, 32, pp. 12363–12379, 2020.

[8] F. Gu, K. Li and Y. Liu, "A Hybrid Evolutionary Algorithm for Solving Function Optimization Problems," 2016 12th International Conference on Computational Intelligence and Security (CIS), Wuxi, December 16-19, 2016.

[9] A. W. Mohamed, H. Z. Sabry, M. Khorshid, "An alternative differential evolution algorithm for global optimization, " *Journal of Advanced Research*, 3(2), 149–165, 2012.

[10] A. Draa, K. Chettah and H. Talbi, "A compound sinusoidal differential evolution algorithm for continuous optimization," *Swarm and Evolutionary Computation*, 50, pp. 100450, 2019.

[11] L. Deng, P. Yang and W. Liu, "An Improved Genetic Algorithm," 2019 IEEE 5th International Conference on Computer and Communications (ICCC), Chengdu, China, December 06-09, 2019.

[12] A. Lambora, K. Gupta and K. Chopra, "Genetic Algorithm-A Literature Review," 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-Con), Faridabad, India, Feb. 14-16, 2019.

[13] D. E. Goldberg, *Genetic Algorithms in Search Optimization and Machine Learning*. Boston: Addison–Wesley Publishing Company, 1989.

[14] J. Spall, "An overview of the simultaneous perturbation method for efficient optimization," *Johns Hopkins APL Technical Digest*, 19: 482-492, 1998.

[15] X. S. Yang, *Computational Optimization*. Methods and Algorithms, 2011.

[16] M. Jamil & X. S. Yang, "A Literature Survey of Benchmark Functions For Global Optimization Problems," *International Journal of Mathematical Modelling and Numerical Optimisation*, 4, 2014.

[17] S. Pietro, M. W. Oliveto, T. Weise, B. Wróbel and A. Zamuda, "Black-Box Discrete Optimization Benchmarking," Workshop at the 2018 Genetic and Evo. Computation Conference (GECCO), Japan, July 15-19, 2018.

[18] H. Eren, "Mathematical Methods of Optimisation," *Handbook of Measuring System Design*, volume 3, 452-455. 2005.

[19] J. M. Powers and M. Sen, *Mathematical Methods in Engineering*. Cambridge University Press, 2015.

[20] R. Storn and K. Price, "Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces," *Journal of Global Optimization*, 11, 341-359, 1997.

[21] J. Holland, *Adaptation in natural and artificial systems*. University of Michigan Press, Extended new Edition, MIT Press, Cambridge, 1975.

[22] M. Tunay and R. Abiyev, "Hybrid Local Search Based Genetic Algorithm and Its Practical Application," *International Journal of Soft Computing and Engineering*, 5(2):21-27, 2015.

[23] M. Dorigo, V. Maniezzo and A. Colorni, "The ant system: optimisation by a colony of cooperative agents," *IEEE Transactions on System Man Cybernet*, 26, 29-41, 1996.

[24] R. Dhiman and J. S. Priyanka, "Genetic algorithms tuned expert model for detection of epileptic seizures from EEG signatures," *Applied Softcomputing*, 19, 8-17, 2014.

[25] E. Lalla-Ruiz, J. L. González-Velarde, B. Melián-Batista, J. M. Moreno-Vega, "Biased random key genetic algorithm for the tactical berth allocation problem," *Applied Softcomputing*, 22, 60-76, 2014.

[26] M. A. Belal and M. H. Haggag, "A structured-population genetic-algorithm based on hierarchical hypercube of genes expressions," *International Journal of Computer Applications*, 64(22), 5-18, 2013.

[27] S. H. Chen, M. C. Chen and Y. C. Liou, Artificial chromosomes with genetic algorithm 2 (ACGA2) for single machine scheduling problems with sequence-dependent setup times," *Applied Softcomputing*, 17, 167-175, 2014.

[28] A. J. Delima, A. Sison and R. Medina, "A modified genetic algorithm with a new crossover mating scheme," *Indonesian Journal of Electrical Engineering and Informatics*, 7, pp. 165-181, 2019.

[29] D. Chaudhary, A. K. Tailor, V. P. Sharma and S. Chaturvedi, "HyGADE: Hybrid of Genetic Algorithm and Differential Evolution Algorithm," 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, July 6-8, 2019.

[30] H. F. Sulaiman, B. T. Sartana and U. Budiyanto, "Genetic Algorithm With Random Crossover and Dynamic Mutation on Bin Packing Problem," 2019 6th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), Bandung, Indonesia, September 18-20, 2019.

[31] N. S. Utami, A. Jazidie and R. E. A. Kadier, "Path Planning for Differential Drive Mobile Robot to Avoid Static Obstacles Collision using Modified Crossover Genetic Algorithm," 2019 International Seminar on Intelligent Technology and Its Applications (ISITIA), Surabaya, Indonesia, August 28-29, 2019.

[32] W. A. H. M. Ghanem and A. A. Jantan, "Cognitively Inspired Hybridization of Artificial Bee Colony and Dragonfly Algorithms for Training Multi-layer Perceptrons," *Cogn Comput*. 10, 1096–1134, 2018.

[33] H. Zhang, S. Li and X. Liu, "Research on Function Optimization Based on Improved Genetic Particle Swarm Optimization," *Journal of Physics: Conference Series*, 1549, 042133, 2020.

