

NEURAL NETWORK WORLD: A NEURAL NETWORK BASED SELECTION METHOD FOR GENETIC ALGORITHMS

Can Yalkın*, Emin Erkan Korkmaz†

Abstract: Genetic algorithms (GAs) are stochastic methods that are widely used in search and optimization. The breeding process is the main driving mechanism for GAs that leads the way to find the global optimum. And the initial phase of the breeding process starts with parent selection. The selection utilized in a GA is effective on the convergence speed of the algorithm. A GA can use different selection mechanisms for choosing parents from the population and in many applications the process generally depends on the fitness values of the individuals. Artificial neural networks (ANNs) are used to decide the appropriate parents by the new hybrid algorithm proposed in this study. And the use of neural networks aims to produce better offspring during the GA search. The neural network utilized in this algorithm tries to learn the structural patterns and correlations that enable two parents to produce high-fit offspring. In the breeding process, the first parent is selected based on the fitness value as usual. Then it is the neural network that decides the appropriate mate for the first parent chosen. Hence, the selection mechanism is not solely dependent on the fitness values in this study. The algorithm is tested with seven benchmark functions. It is observed from results of these tests that the new selection method leads genetic algorithm to converge faster.

Key words: *Genetic algorithms, neural networks, selection, hybrid algorithms*

Received: October 16, 2011

Revised and accepted: October 15, 2012

1. Introduction

The origin of genetic algorithms (GAs) comes from the principle of genetics and evolution. The algorithmic framework was first developed by John Holland [9]. GAs were introduced as a tool for optimization problems. The algorithm uses a population of individuals for the search process. The strategy is applicable to

*Can Yalkın

Department of Computer Engineering, Yeditepe University, Istanbul, Turkey, Tel.: +90(533)2408368, E-mail: canyalkin@gmail.com

†Emin Erkan Korkmaz

Department of Computer Engineering, Yeditepe University, Istanbul, Turkey, Tel.: +90(216)5780426, E-mail: ekorkmaz@cse.yeditepe.edu.tr

different kinds of problems. GAs do not guarantee to find the global optimum solution, however they usually provide a satisfactory solution within the tolerance limits [18], [23].

GAs are powerful techniques for search and optimization, however they have certain drawbacks. In some domains, it can take a very long time for the GA to converge to the optimal solution. GA operators might be insufficient to provide fine tuning that will give the exact optimum solution; even when GAs can approach global optimum area. In order to overcome this situation, GAs are supported with local search methods. With the use of local search, GA can enhance the ability of local exploitation and thanks to which the solution can be found faster.

Due to this fact, GAs and local search techniques can be considered as complementary of each other. The hybridization of GAs with local search methods yield satisfactory results in many domains [6]. Generally, hybrid methods combine more than one intelligent technique to solve challenging problems.

An artificial neural network (ANN) imitates the brain's nervous system which is made up of neurons. An ANN can both be used to solve classification problems and approximate any non-linear functions. They have a strong capability of learning from samples and making predictions [7]. ANN can be a good supplement for the hybrid methods that need predictions.

It is possible to find different studies that use GAs to determine the structure and initial weights of neural networks. In this study, we propose a hybrid algorithm which uses neural networks to predict the parent pairs that will hopefully create better offspring during the GA search. The algorithm is tested with seven benchmark functions and the results are compared with the results of the simple genetic algorithm and a genetic algorithm that uses a special selection process (OSGA) [1]. It can be denoted from the results that the neural network utilized in the algorithm can learn the patterns in the parent pairs that increase the potential of producing better offspring. The hybrid algorithm proposed in this study outperforms both the simple GA and OSGA.

2. Related Work

There are many researches which combine genetic algorithms and artificial neural networks. In most of the proposed studies, GAs are used either for improving the learning rate of ANNs or designing network architecture.

[13] is a study that uses ANN in the framework of GAs. The neural network is used to guess combination of design variables that is expected to have a better fitness value than the worst chromosome in the current generation. If the neural network can make a successful guess, then the predicted chromosome is replaced by the worst chromosome in the current generation. There are also other studies where neural networks are used for function approximation instead of complex and time-consuming fitness calculations [26], [10], [24], [5].

The architecture of neural networks cannot be easily constructed for the given problem. In [20] a method is proposed to evolve neural network architectures using genetic algorithms. The study covers a new crossover method and a pruning strategy that aims to improve network architecture, activation function and weights in the network. In [8], [22] genetic algorithms are also used to create an appropriate

network structure. There are also studies where genetic algorithms are used instead of back propagation learning algorithm [4], [19], [25].

Besides hybrid algorithms, there are also researches on the selection mechanism in GAs. The proposed approaches aim to improve population diversity and prevent convergence to local optimum. For instance, [16] offers two new types of selection mechanisms to sustain population diversity. The new selection mechanisms take into account correlations between individuals. The first type, namely correlative tournament selection, chooses the first parent by using conventional tournament selection but the second parent is chosen according to its fitness value and Hamming distance to the first parent. The one with a high fitness and a smaller Hamming distance is chosen to be the second parent. The second type of selection mechanism determines the individuals that will be transferred to the next generation. The first element that will be transferred is the individual with the highest fitness among the two parents and the two offsprings produced. The second individual is chosen by evaluating the Hamming distance between the first element chosen and remaining three individuals. The individual, which has the highest value of Hamming distance, is selected as the second.

In [1] authors propose a method called offspring selection (OSGA) which takes place after the reproduction step. The purpose of this method is to prevent premature convergence or at least to reduce it by preserving genetic diversity. GA passes its reproduction phase in a conventional way. Later on, in order to accomplish the proposed objective, this method takes into account a comparison of fitness value of the produced offspring and its parents' after the reproduction phase. If the produced offspring has superior fitness than its parents, then it is accepted for the next generation. The offspring has superior fitness, if its fitness value exceeds both parents' fitness value. However, at the beginning of the search process, the offspring is accepted to be superior even when it's better than only one parent. Therefore, what this method ensures is that in each generation there exists a number of better individuals compared to the previous generation.

In study [2] the proposal is about non-random mating and varying population size for GA. The study works out the effects of non-incest GA (niGAVaPS) with varying population size and negative assortative mating GA with varying population size (nAMGAVaPS). The niGAVaPS uses incest prohibition method to provide non-random mating. In this method, each individual has an ancestry table and according to a predefined value it is forbidden for the individuals to mate with these in the ancestry table. The nAMGAVaPS method chooses first parent and a set of individuals with one of the conventional selection methods. For an individual to become the second parent, it should have the largest hamming distance to the first parent. Varying population size is obtained by the life time calculation. Each individual has a life time which indicates how long an individual is kept in population and it is calculated according to fitness value of the individual. And better individuals, who have higher fitness values, reside in the population more than the others and they are more likely to contribute to the reproduction phase. In other words, better individuals reside in the population more than the others and they have more chance to contribute to reproduction phase.

In [3] a new selection method is introduced for GA. Algorithm sorts the chromosomes in the population according to their fitness values and then divides the

population into two parts which are called high fit and low fit. It is specifically emphasized that the individual to be chosen from high fit has to be the first individual whereas the individual to be picked up among the low fit can be any. When a new offspring is formed, it is appended to the population. Then population is sorted again and for the size of the population to be the same as before, the population is lessened from the tail. This cycle is processed until termination criteria is met.

Study [11] proposes a selection method which is not dependent only on high fit individuals unlike standard selection methods. In other words, the aim of this study is to create better selection pressure and to preserve genetic diversity. The method first chooses a fitness value f , uniformly from a set of fitness values from minimum to maximum value. Then an individual, which has fitness f , is selected randomly from population and a copy of this individual is added to the population after mutation and recombination. It is claimed that the proposed selection mechanism is more effective than standard selection methods.

3. Genetic Algorithms

GAs are stochastic methods inspired from natural evolution and they are useful in the scope of search and optimization. GAs are known to be robust and efficient. The search process is carried out on a population of individuals in GAs. This enables GAs to find the global optimum even in complex search spaces. The framework proposed by GAs is flexible and can be applied to different problem domains. GAs start the search process with a randomly generated population of individuals. Genetic operators such as selection, crossover and mutation are used to breed new generations. Each individual is assigned a fitness value that defines the quality of that individual. Selection is the process of choosing parents from population for reproduction. There are various selection methods, but most of them share the common idea that well fit individuals have more chance to be selected. Roulette wheel, tournament, rank selection are some examples for the process. Selection is an important process, since it determines the direction of the search and hence it effects the convergence speed. Selection operation can be considered to be the main reason for premature convergence in many cases. The main driving force of GAs is the interaction between selection and crossover. In other words, for the crossover operation, that is the process of combining two parents and generating offspring as the result, to become successful, the selection mechanism should find coherent parents. The main idea of crossover operation is to obtain offspring with better fitness values. Using only crossover operation during the search is not sufficient. Mutations are also introduced both to prevent premature convergence and to provide diversity. It is important to keep the diversity in the population. Otherwise, the effect of genetic operators diminish and it becomes impossible to converge to the optimal solution [9], [17], [23].

4. Neural Networks

An artificial neural network (ANN) is a structure composed of small computational units named as artificial neurons. A distributed computation is carried out by these

units in an ANN. The neurons are placed to the network in layers. In each neuron, the input data are transferred to the other neurons via connection links, after they are processed by the neuron. Each neuron has an activation function that is applied to the sum of its inputs to determine corresponding output of that neuron. Hence, the activation function is applied to the input data as follows.

$$a_i = g \left(\sum_{j=0}^n w_{j,i} a_j \right). \quad (1)$$

There are a few types of activation functions but generally non-linear and sigmoid activation functions are used. Sigmoid function is

$$f(x) = \frac{1}{1 + e^{-\sigma x}}. \quad (2)$$

There are two types of structures for ANNs; feed forward networks and recurrent networks. Feedforward networks carry data from input to its output whereas recurrent networks have also connections to inputs from the outputs of the neurons. Besides network architecture, training is crucial for a neural network, as the weights of the neurons are adjusted during the training process. The weights have to be adjusted to the correct values in order to obtain a successful coverage over the training data [15], [14].

Constructing the architecture of neural networks is a difficult issue. There is no straightforward answer to the question of how many layers an ANN should have for a given problem. However, it is known that ANNs can represent any non-linear function if a sufficient number of neurons and a hidden layer exist in the network [7].

In this study, a feed forward network with three layers is utilized. The threshold function of the neurons is bipolar sigmoid function and Rprop learning method is used during the training [12], [21]. Rprop learning method is a modified version of back propagation algorithm and it converges faster compared to the standard back propagation method. Also, Rprop learning reduces the effects of initialization and the learning rate is adaptive in this algorithm.

5. Hybrid Genetic Algorithm

In this study, a new selection method based on ANNs is proposed for GAs. An ANN is used to determine whether the two selected parents have the potential to produce high-fit offspring or not. The conventional selection mechanisms use fitness values to determine the parents that would be chosen for the crossover operation. High-fit individuals have more chance to be selected as a parent. However, this scheme does not always operate as expected. Even if the parents have high fitness values, it might be more probable that a bad schema would appear in the offspring, if the individuals chosen are not correlated in a particular way. Such situations can lead to premature convergence or they at least slow down the convergence speed of the GA utilized.

In this study, a new method is used to decide whether the recombination of two chromosomes would produce a high-fit offspring. An ANN is utilized to analyze

the structural properties of the chromosomes that are likely to produce high-fit offspring. The ANN is expected to learn if the chosen parents have appropriate and coherent building blocks that would combine in the offspring to increase its fitness value. The standard crossover operation is a stochastic process. A certain percentage of the offspring produced by this operation is expected to have better fitness values compared to their parent chromosomes. This happens when the building blocks of the parents combine coherently in the offspring chromosome. Certainly, when ANNs are utilized for the selection process, the ANN cannot guarantee to provide such high-fit offspring in all of the crossover operations; however the learning process that takes place in the ANN is expected to increase the ratio of obtaining better offspring during the recombination process. When this is achieved, the GA search is expected to converge faster than the standard GA framework.

The proposed GA starts with the standard approach by initializing the population with randomly created chromosomes and then new generations are bred using the standard crossover and mutation operations. The standard tournament selection is used to determine the parents for crossover in this phase. On the other side, while this standard GA search is carried out, it is needed to collect training data for the ANN. The data collection phase continues up to a certain predefined generation. Two new parameters are introduced which are called sample per generation ($spg \in [1, P]$, where P denotes the population size) and training parameter (TP). The two parents used in the crossover operation and the fitness of the corresponding offspring form an element of the training set. And fitness values here are constructed with the objective function. At each generation, spg number of parent pairs are randomly chosen and they are added to the training set; together with the corresponding offspring fitness values.

When generation number reaches to TP value, training data are sorted with respect to fitness value and the fitness values are normalized between -1 and 1 . The ANN is trained with these data. After training, the GA switches from ordinary tournament selection to the new selection method. In this new approach, parent chromosomes are fed to the ANN and an output value between -1 and 1 is obtained. When the ANN predicts that the input parents are likely to create successful offspring, the output of the network is closer to 1 . The value -1 is the case where the parents do not form a coherent pair to produce a high-fit offspring.

In fact, the first parent is chosen again using tournament selection in this new framework. Then the ANN is used to determine an appropriate mate for the first parent chosen.

Let p_1 be the first parent such that

$$p_1 = \min(f(p_j)) \text{ for } j = 1 \dots n, \quad (3)$$

where f is the objective function, p_j is an element in the tournament set T which is composed of randomly chosen elements from the population. Also n is the tournament size.

Second parent is chosen by using the ANN as follows. Let $A = \bigcup_{j=1}^n p_j$, be a randomly formed set from the population ($p_j \in P$). The size of set A is equal to the tournament set (T) size in this study. The second parent p_2 is chosen using the following equation.

$$p_2 = \max(ANN(p_1, p_j)), \quad (4)$$

where $p_j \in A$. Hence, each individual in set A is queried in the ANN to determine if it is a coherent mate with the primarily chosen parent p_1 . Then the individual that has the highest score from the ANN is chosen as the second parent. Algorithm 1 presents the pseudocode for the new hybrid selection algorithm proposed in this study.

The success of the newly proposed method is based on how well the ANN could be trained on the chosen parent pairs and the offspring they produce. A sample case can be considered with the sphere function. Sphere function is a function that sums the square of variables and its optimization criterion is minimization. ($f(x) = \sum_{i=1}^D (x_i)^2$). Sphere function gets minimum value when all variables are zero. Therefore, chromosomes with more zero values in the genes would have better fitness values compared to the others. However, the recombination of two parents may not produce better offspring if the parents have the zero values in similar genes. In this case, the ANN can induce that, if the parents chosen for crossover, have zero values in different parts of the chromosomes (i.e their hamming distance is large), then the chance of producing better offspring increases. Hence, the trained ANN could be utilized to determine the appropriate mate for a chosen parent.

Algorithm 1 Genetic Algorithm with neural network

```

createRandomGeneration()
currentGeneration=1
while currentGeneration < finalGeneration do
  if currentGeneration < TP then
    tournamentSelection()
    takeSampleFromPopulation()
  else
    if currentGeneration==TP then
      trainNN()
    end if
    useNNSelection()
  end if
  crossover()
  mutation()
  currentGeneration++
end while

```

6. Experimental Results

6.1 Functions and parameters

The proposed algorithm is tested by using seven benchmark functions and the results are compared with the results of standard genetic algorithm. Additionally, a comparison is carried out with OSGA [1] on the same function set. As stated

in Section 2, OSGA is a genetic algorithm which uses a special selection process. Hence, the performance of the proposed method has been evaluated based on another GA that also focuses on the selection process. The benchmark functions are taken from [27] and [17]. All tests are executed with the same parameters such that crossover probability is 0.85, mutation probability is 0.005, population size is 60 and 1000 generations is used in each run. Binary encoding has been preferred since it is a commonly used encoding in the literature. Moreover, 12 bits are used to represent a variable. 50 runs are utilized for each test to obtain reliable and accurate results. The same framework is also used in OSGA runs. A neural network with three layers is used in the hybrid framework. The number of neurons in the input layer is two times the chromosome length. Five neurons exist in the hidden layer and a single neuron in the output layer. *Rprop learning* [12], [21] algorithm is used with 3000 epochs due to its fast learning ability.

The functions used in the experiments are listed below.

Generalized Rastrigin Function:

$$f_{Ra}(x) = \sum_{i=1}^D [x_i^2 - 10\cos(2\pi x_i) + 10], \quad (5)$$

where $-5.12 \leq x_i \leq 5.12$.

Generalized Schwefel Function:

$$f_{Sc}(x) = - \sum_{i=1}^D x_i \sin(\sqrt{|x_i|}), \quad (6)$$

where $-500 \leq x_i \leq 500$.

Generalized Rosenbrock Function:

$$f_{Ro}(x) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2], \quad (7)$$

where $-29 \leq x_i \leq 31$.

Shubert Function:

$$f_{Sh}(x) = \sum_{i=1}^5 i \cos[(i+1)x_1 + i] \cdot \sum_{i=1}^5 i \cos[(i+1)x_2 + i], \quad (8)$$

where $-10 \leq x_i \leq 10$ for $i = 1, 2$.

Shekel's Foxholes Function:

$$f_{Shk}(x) = \left[\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{i,j})^6} \right]^{-1}, \quad (9)$$

where $-98 \leq x_i \leq 34$ for $i = 1, 2$ and

$$a_{i,j} = \begin{bmatrix} -32 & -16 & 0 & 16 & 32 & -32 & \dots & 0 & 16 & 32 \\ -32 & -32 & -32 & -32 & -32 & -16 & \dots & 32 & 32 & 32 \end{bmatrix},$$

Colville Function:

$$\begin{aligned} f_{Co}(x) = & 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + \\ & + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 \\ & + 10.1((x_2 - 1)^2 + (x_4 - 1)^2) \\ & + 19.8(x_2 - 1)(x_4 - 1), \end{aligned} \quad (10)$$

where $-10 \leq x_i \leq 10$ for $i = 1, 2, 3, 4$.

Generalized Griewank Function:

$$f_{Gr}(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, \quad (11)$$

where $-600 \leq x_i \leq 600$.

Function 5 is one of DeJong's functions. Due to the cosine term, it has a number of local minima that are regularly distributed and one global optimum $f_{Ra}(x) = 0$ at $x = [0, 0, \dots, 0]$. The function is used with 3 variables for testing the hybrid algorithm.

Function 6 has many local minima and one global optimum. The function space is complicated, where optimization algorithms may stuck on the local optimum. The global minimum of the function is $f_{Sc}(x) = -418,9829N$, (where N is a variable number) at the point $x = [420.9687, \dots, 420.9687]$. The function is used with 3 variables in the tests.

Function 7 is also named as banana function. The function has only one global minimum $f_{Ro}(x) = 0$ at $x = [1, \dots, 1]$ in the long paraboloid function space. Although the space formed by the function is not complicated, convergence to the global optimum is difficult. The function is used with 3 variables in the tests.

Function 8 has many local minima (742 local minima and 18 global minima) $f_{Sh}(x) = -186.73$. The function is used with 2 variables in the tests.

Function 9 has a global minimum of $f_{Shk}(x) \cong 1$ at $x = [-32, -32]$. The function is used with 2 variables in the experiments.

Function 10 has a global minimum $f_{Co}(x) = 0$ at $x = [1, 1, 1, 1]$. The function is used with 4 variables in the tests.

Function 11 is one of the most beneficial benchmark functions for testing genetic algorithms. It has a number of local minima which are proportional to dimension number. Function has global minimum $f_{Gr}(x) = 0$ at $x = [0, \dots, 0]$ and it is used with 3 variables in the tests.

6.2 Results

Since the Offspring selection genetic algorithm (OSGA) tries to create better offspring than its parents' in every generation, OSGA converges to the global optimum area better than the SGA. Hence, it is essential to compare OSGA and

GANN as they both outperform the SGA. OSGA and GANN results are presented in Tab. I and II. In the Table, average fitness and standard deviation of seven functions are shown both for OSGA and GANN. According to the Table, it is obviously seen that GANN has better results in terms of average fitness values compared to OSGA except Schwefel function. Also, again excluding Schwefel function, standard deviation is smaller and this can be evaluated as an indicator of stability.

The results of the SGA and GANN experiments are presented in Fig. 1. In this figure, average best fitness value and the corresponding standard deviation is given for all experiments. For each experiment, the average value is obtained from 50 runs with different random seeds. In each chart, the results obtained by the standard GA framework and the newly proposed hybrid algorithm are compared on a benchmark function. SGA represents simple genetic algorithm and GANN represents the new method. It is obvious that for each benchmark function, GANN has improved the solution quality in terms of average best fitness. Furthermore, the hybrid framework has a smaller standard deviation for all of the benchmark functions

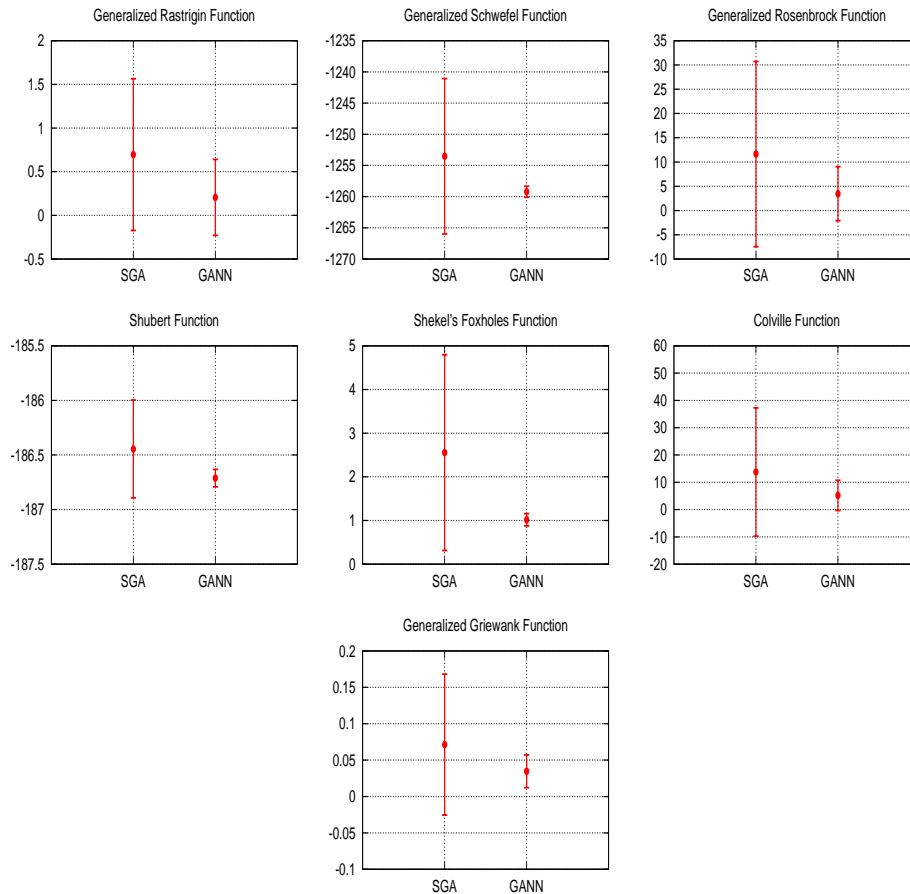


Fig. 1 Average fitness and standard deviation of each benchmark function.

Gen. Rastrigin Func.		Gen. Schwefel Func.		Gen. Rosenbrock Func.		Shubert Func.	
OSGA	GANN	OSGA	GANN	OSGA	GANN	OSGA	GANN
Avg.	0.50	-1258.64	-1256.98	43.88	19.48	-185.54	-186.72
Std.	0.64	4.54	16.72	160.87	109.40	6.19	0.014
Dev.							

Tab. I Average fitness and standard deviation of OSGA and GANN.

Shekel's Foxholes Func.		Colville Func.		Gen. Griewank Func.	
OSGA	GANN	OSGA	GANN	OSGA	GANN
Avg.	2.31	21.67	6.10	0.03	0.03
Std.	2.15	47.56	12.11	0.0245	0.0244
Dev.					

Tab. II Average fitness and standard deviation of OSGA and GANN cont.

Generalized Rastrigin Function		Generalized Schwefel Function		Generalized Rosenbrock Function		Shubert Function		Shekel's Foxholes Function		Colville Function		Generalized Griewank Function	
0.000567	0.001723	0.004524	0.000074	0.000005	0.01376315	0.01038109							

Tab. III T-test for benchmark functions.

used. It can be stated that the hybrid framework provides a more robust search compared to SGA. On the other side, statistical tests are needed to guarantee that the improvement obtained with the new methodology is statistically significant. Consequently, *t-test* is applied for each experiment carried out. Tab. III presents the results obtained in this statistical test.

According to Tab. III, in all experiments, the *t-test* values are below 0.05. Hence, the hybrid framework introduced a statistically significant improvement for all of the benchmark functions used in the tests.

The results presented above, provide information about the contribution of ANNs to the performance of the genetic search. In order to provide more insight about this contribution, the percentage of successful crossover operations both for SGA and GANN are presented in Tab. IV. A crossover operation is considered to be successful, if the offspring produced is superior to one or both of the parents. As it can be seen in the Table, the percentage of successful operations in GANN runs is very high compared to SGA. Therefore, it can be deduced that neural network mechanism has an impulsive force for the selection phase that leads GA to obtain better solutions and smaller deviations.

Tab. V presents learning ratio of the neural networks used in the GANN. 10-fold cross validation method is used to obtain classification rates and it can be mentioned that neural network can learn the patterns in the training set without overfitting the data. Consequently, GA supported with this selection process based on ANNs can achieve better results and smaller deviations.

The hybrid algorithm uses a neural network for the selection mechanism. A data collection and a learning process take place in the hybrid framework. Hence, the ANN utilized in the hybrid algorithm consumes extra computational time compared to simple genetic algorithm. The performance of the simple genetic algorithm is analyzed when it is also allowed to use the extra computational time needed by the ANN in the hybrid framework. Consequently, the generation number of the simple genetic algorithm is extended such that the two methodologies use exactly the same computational time.

Fig. 2 again presents the average best fitness values and the corresponding standard deviations. Here, the simple genetic algorithm uses some extra generations that corresponds to the extra computational time used by the ANN in the hybrid framework. In terms of average best fitness the new methodology is still better and the standard deviation is still smaller compared to the simple genetic algorithm. However, in the *t-test*, the significance is lost for Schwefel and Griewank functions. For the remaining functions, the improvement obtained by the hybrid framework is still statistically significant. Tab. VI represents the results obtained in the *t-test*.

To sum up, genetic algorithm with the new selection method has better results. The hybrid method chooses parents for crossover according to the feedback obtained from the ANN and it converges faster compared to SGA and OSGA. It can be proposed that, the neural network can learn the regularities in the building blocks that make two chromosomes as appropriate mates for each other. The success of the learning process is validated on different benchmark functions with different structural properties.

	Gen. Rastrigin Function	Gen. Schwefel Function	Gen. Rosenbrock Function	Shubert Function	Shekel's Foxholes Function	Colville Function	Gen. Griewank Function
SGA	2%	2%	2%	1%	1%	5%	2%
GANN	76%	79%	77%	76%	77%	77%	76%

Tab. IV Success Ratios of SGA and GANN.

	Gen. Rastrigin Function	Gen. Schwefel Function	Gen. Rosenbrock Function	Shubert Function	Shekel's Foxholes Function	Colville Function	Gen. Griewank Function
Corr. Classified	88.33%	89.29%	82.14%	75%	72.5%	91.75%	71.21%
Incorr. Classified	11.67%	10.71%	17.86%	25%	27.50%	8.25%	28.79%

Tab. V Neural Network 10 fold cross validation results.

	Generalized Rastrigin Function	Generalized Schwefel Function	Generalized Rosenbrock Function	Shubert Function	Shekel's Foxholes Function	Colville Function	Generalized Griewank Function
0.034708	0.396558	0.004923	0.000350	0.000007	0.047916	0.304591	

Tab. VI T-test for benchmark functions.

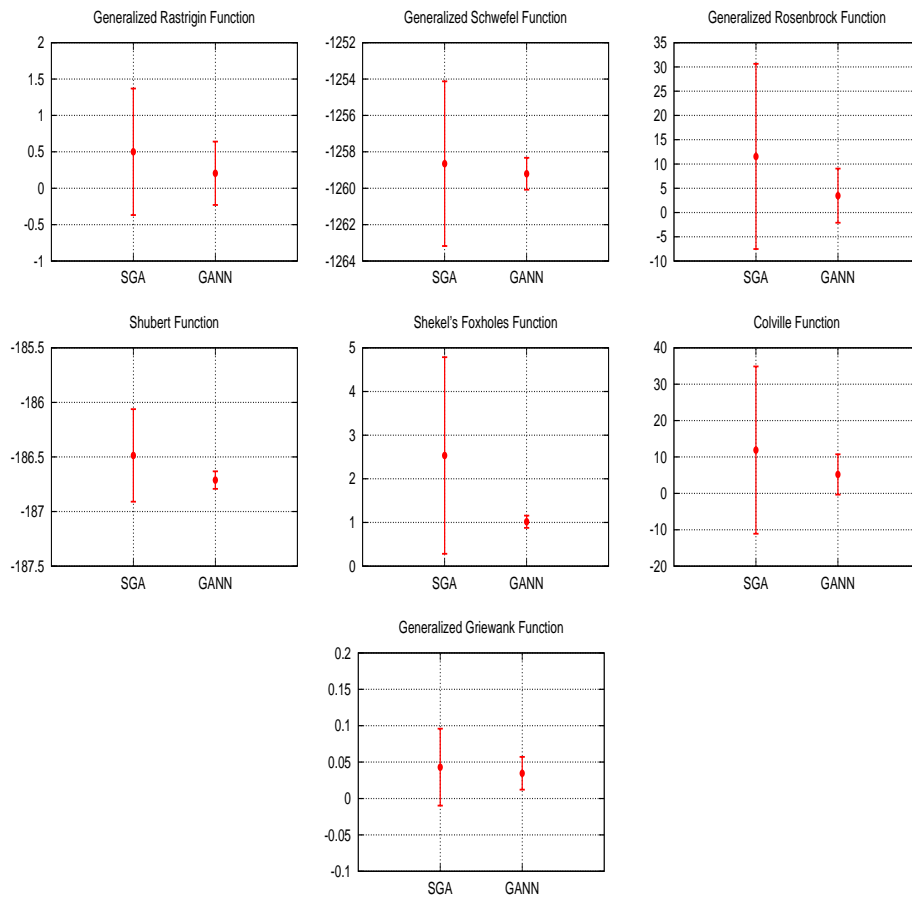


Fig. 2 Average fitness and standard deviation of each benchmark function with extended generation.

7. Conclusion

In this paper, we have presented a new hybrid algorithm that utilizes neural networks in the selection process of genetic algorithms. The neural network used in the hybrid framework learns the patterns and correlations that increase the potential to produce better offspring. The experiments carried out on seven different benchmark functions denote that the hybrid method has a significantly better convergence curve. Therefore, it can be claimed that the neural network can successfully learn the structural patterns and regularities in the chromosomes that can be used to improve the success of reproduction.

The methodology is tested on the benchmark functions that are widely used for testing genetic algorithms. As subsequent work, this hybrid mechanism can be examined on more complex real-world problems. The contribution of the methodology could be more clear when the framework is tested in areas like combinatorial optimization where simple genetic algorithms face serious convergence problems.

References

- [1] Affenzeller M., Wagner S.: Offspring selection: A new self-adaptive selection scheme for genetic algorithms. *Adaptive and Natural Computing Algorithms*, Springer Computer Science, 2005, pp. 218–221.
- [2] Agostinho C. F., Fernandes C., Rosa A.: A study on non-random mating and varying population size in genetic algorithms using a royal road function. In: *IEEE CECA01, Proc. of the 2001 IEEE Congress on Evolutionary Computation*, IEEE Press, 2001, pp. 60–66.
- [3] Ali E. E. E.: A proposed genetic algorithm selection method. King Saud University, College of Computer and Information Sciences, 2006.
- [4] Bornholdt S., Graudenz D.: Original contribution: General asymmetric neural networks and structure design by genetic algorithms. *Neural Netw.*, **5**, 2, Feb. 1992, pp. 327–334.
- [5] Cheng J., Li Q.: Reliability analysis of structures using artificial neural network based genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*, 197, Aug. 2008, pp. 3742–3750.
- [6] El-Mihoub T. A., Hopgood A. A., Nolle L., Battersby A.: Hybrid genetic algorithms: A review. *Engineering Letters*, 13, 2006, pp. 124–137.
- [7] Fausett L.: *Fundamentals of Neural Networks : Architectures, Algorithms, and Applications*. Prentice Hall, 1994.
- [8] Ferentinos K. P.: Biological engineering applications of feedforward neural networks designed and parameterized by genetic algorithms. *Neural Netw.*, **18**, 7, Sept. 2005, pp. 934–950.
- [9] Goldberg D. E.: *Genetic Algorithms in Search, Optimization & Machine Learning*. Addison Wesley, Inc., 2006.
- [10] Gozalves J. M., Yilmaz S., Alim F., Ivanov K., Levine S. H.: Sensitivity study on determining an efficient set of fuel assembly parameters in training data for designing of neural networks in hybrid genetic algorithms. *Annals of Nuclear Energy*, 33, Mar. 2006, pp. 457–465.
- [11] Hutter M.: Fitness uniform selection to preserve genetic diversity. In: *In Proc. 2002 Congress on Evolutionary Computation (CEC-2002, IEEE, 2002)*, pp. 783–788.
- [12] Igel C., Hüsken M.: Improving the rprop learning algorithm. In: *Proceedings of the Second International Symposium on Neural Computation, NC 2000, ICSC Academic Press, 2000*, pp. 115–121.
- [13] Javadi A., Farmani R., Tan T.: A hybrid intelligent genetic algorithm. *Advanced Engineering Informatics*, 19, Oct. 2005, pp. 255–262.
- [14] LeCun Y.: A theoretical framework for back-propagation. In: *Proceedings of the 1988 Connectionist Models Summer School (CMU, Pittsburgh, Pa, 1988)*, D. Touretzky, G. Hinton, and T. Sejnowski, Eds., Morgan Kaufmann, pp. 21–28.
- [15] LeCun Y., Bottou L., Orr G., Muller K.: Efficient backprop. In: *Neural Networks: Tricks of the trade (1998)*, G. Orr and M. K., Eds., Springer.
- [16] Matsui K.: New selection method to improve the population diversity in genetic algorithms. *Systems, Man, and Cybernetics*, 1999. *IEEE SMC '99 Conference Proceedings. 1999 IEEE International Conference*, 1, Oct. 1999.
- [17] Michalewicz Z.: *Genetic Algorithms + Data Structures = Evolution Programs*, 3 ed. Springer, 1992.
- [18] Mitchell M.: *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA, USA, 1996.
- [19] Orozco-Montegudo M., Taboada-Crispí A., Del Toro-Almenares A.: Training of multi-layer perceptron neural networks by using cellular genetic algorithms. In: *Proceedings of the 11th Iberoamerican conference on Progress in Pattern Recognition, Image Analysis and Applications (Berlin, Heidelberg, 2006), CIARP'06, Springer-Verlag*, pp. 389–398.
- [20] Pujol J. C. F., Poli R.: Evolving neural networks using a dual representation with a combined crossover operator. *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence, 1998*, pp. 416–421.

- [21] Riedmiller M.: Rprop – description and implementation details, 1994.
- [22] Saemi M., Ahmadi M., Varjani A. Y.: Design of neural networks using genetic algorithm for the permeability estimation of the reservoir. *Journal of Petroleum Science and Engineering*, 59, Oct. 2007, pp. 97–105.
- [23] Sivanandam S. N., Deepa S. N.: Introduction to Genetic Algorithms, 1st ed. Springer Publishing Company, Incorporated, 2007.
- [24] Tse D. C., Chan L. Y.: Application of micro genetic algorithms and neural networks for airfoil design optimization. RTO MP-035 Aerodynamic Design and Optimization of Flight Vehicles in a Concurrent Multi-Disciplinary Environment, 1999.
- [25] Wang G. S., Huang F.-K.: Ga-based neural network to identification of nonlinear structural systems. In: Proceedings of the 4th International Symposium on Neural Networks: Advances in Neural Networks, Part III, (Berlin, Heidelberg, 2007), ISNN'07, Springer-Verlag, pp. 57–65.
- [26] Wang L.: A hybrid genetic algorithm–neural network strategy for simulation optimization. *Applied Mathematics and Computation*, 170, Nov. 2005, pp. 1329–1343.
- [27] Yuen S. Y., Chow C. K.: A genetic algorithm that adaptively mutates and never revisits. *IEEE Transactions On Evolutionary Computation*, 13, Apr. 2009, pp. 454–472.