

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/328367728>

# Genetic Algorithm Based on Enhanced Selection and Log-Scaled Mutation Technique: Volume 1

Chapter · January 2019

DOI: 10.1007/978-3-030-02686-8\_55

CITATIONS

23

READS

654

4 authors:



**Neeraj Gupta**

Oakland University

77 PUBLICATIONS 783 CITATIONS

[SEE PROFILE](#)



**Nilesh Patel**

Oakland University

86 PUBLICATIONS 1,194 CITATIONS

[SEE PROFILE](#)



**Bhupendra Nath Tiwari**

INFN - Istituto Nazionale di Fisica Nucleare

83 PUBLICATIONS 512 CITATIONS

[SEE PROFILE](#)



**Mahdi Khosravy**

Cross Labs

108 PUBLICATIONS 901 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Unification of fundamental interactions [View project](#)



Engineering & Technology: Optimally self driven vehicles [View project](#)

# Genetic Algorithm based on Enhanced Selection and Log-scaled Mutation Technique

Neeraj Gupta<sup>1</sup>, Nilesh Patel<sup>1</sup>, Bhupendra Nath Tiwari<sup>2</sup>, and Mahdi Khosravy<sup>3</sup>

<sup>1</sup> Dept. of Computer Science and Engineering, Oakland University, MI, USA

<sup>2</sup> INFN-Laboratori Nazionali di Frascati, Via. E. Fermi, 40 – I – 00044 Frascati, Rome, Italy

<sup>3</sup> Dept. of Electrical and Electronics Engineering, Fedral University of Juiz de Fora, Brazil

neerajgupta@oakland.edu, npatel@oakland.edu,

bhupendray2.tiwari.phd@iitkalumni.org, mahdi.khosravy@ufjf.edu.br

**Abstract.** In this paper, we introduce the selection and mutation schemes to enhance the computational power of Genetic Algorithm (GA) for global optimization of multi-modal problems. Proposed operators make the GA an efficient optimizer in comparison of other variants of GA with improved precision, consistency and diversity. Due to the presented selection and mutation schemes improved GA, as named Enhanced Selection and Log-scaled Mutation GA (ESA-LOGA), selects the best chromosomes from a pool of parents and children after crossover. Indeed, the proposed GA algorithm is adaptive due to the log-scaled mutation scheme, which corresponds to the fitness of current population at each stage of its execution. Our proposal is further supported via the simulation and comparative analysis with standard GA (SGA) and other variants of GA for a class of multi-variable objective functions. Additionally, comparative results with other optimizers such as Probabilistic Bee Algorithm (PBA), Invasive Weed Optimizer (IWO), and Shuffled Frog Leap Algorithm (SFLA) are presented on higher number of variables to show the effectiveness of ESALOGA.

**Keywords:** Selection operator, Mutation operator, Log-scaled mutation, Diversity preservation, Genetic algorithms, Metropolis algorithm

## 1 Introduction

Rapid industrial growth and utilization of the available resources efficiently are of the prime importance nowadays, for example, route identification in traffic systems, optimization of process allocation in maximizing production, utilization of energy resources in power systems, optimizing VLSI circuits design, CAN optimization in vehicles, etc. [1-12]. Most of the industrial problems are complex in nature and belong to the combinatorial optimization, where the main focus is to optimize discrete variables for maximizing/minimizing required objectives [1, 2]. Although, two methods are available to solve this type of problems, which are integer programming approach and dynamic programming. These traditional methods are known as exact algorithms [3, 5]. However, due to the computational complexity, where a fast solution is required for huge size optimization problems that cannot be relied on such algorithms. Optimization in this aspect may be critically crucial for the sustainable growth of industries in competitively winning and in highly uncertain economic environments [3-5].

Henceforth, in last two decades, as an alternative approach to solve combinatorial problems, a large number of researchers have focused on approximate methods to solve these problems that are close to their optimal state in a reasonably acceptable time. Thus, the development of heuristic algorithms in the field of mathematics, engineering, etc. [6, 7] have demonstrated a successful implementation towards the solution of real-life problems. As a result, a considerable number of heuristic evolutionary algorithms have invented to work efficiently on linear/ nonlinear, differentiable/ non-differentiable, concave/ convex problems with discrete variables [6-9]. A general description of complex functions can be seen in [10] and their applications with discrete variables on power system design in [11, 12], and the capacity of the energy generators, the quantity of goods produced, number of vehicles on the route, etc. in [13, 14, 15].

GA, as it works on binary variables, hardware friendly algorithms have been proposed in many variants to solve the combinatorial problems. The literature-survey shows a huge scope to further improve it by an appropriate combination of mathematical modeling along with the heuristic concept [9]. GA and its associated variants have been proved to give globally optimal solutions, especially for the multi-modal non-differentiable/ combinatorial/ industrial problems [16, 17, 18]. Moreover, GA is very easy to implement and has an advantage of developing its operators in a simple process from the inspiration of genetical processes which have been rigorously investigated at a large scale during the last two decades [1-19].

As developed by John Henry Holland [20], GA is inspired from the “survival of the fittest principle”, which mimics the natural process of evolution in terms of several operators as the selection, crossover, and mutation operators [20]. An adaptation of these operators is analyzed and modeled by a large community of researchers, where several of them have given evidence and they have improved it by introducing novel selection approaches of the fittest individuals, types of crossover variants and mutation schemes. These improved models of GA keep the search not to stuck in a premature convergence. In the light of GA research, this paper offers a combination of mathematical modeling and heuristic approach together in order to find the global optimal solutions for multimodal nonlinear functions. It is worth mentioning that over the last few decades GA has been elected as a successful heuristic evolutionary technique for addressing various global combinational industrial problems and it has been widely used due to its simple structure, see for instance [9, 16, 21-23].

Regardless of the state of affairs, GA has as powerful optimization fundamentals with a few drawbacks that can be seen in a number of readings [8, 9, 16, 24]. GA convergences prematurely due to improper selection, crossover and mutation probabilities and associated criteria [25]. In this paper, a variant of GA has been described as a modification of the GA model parameters, i.e., selection method, crossover operator, mutation operator, and undermining probabilities. Based on the ref. [28], elitism ensures that winner chromosomes go in the next-generation process that moves the search from a premature to the mature phase. This is exploited in Section 3. Hereby, in the light of Adaptive GA [29], our proposal further gives motivations to evolve its mutation probability based on the present state of all candidates by using a probabilistic modeling.

This paper is structured in six sections. Firstly, in Section 2, we provide a brief step-by-step description of the GA algorithm, as our proposal arises its improvement. In Section 3, as the most important part of this paper, a brief description of the proposed enhanced selection scheme and log-scaled mutation operators are provided. Consequently, Section 4 presents a binary coded Enhanced Selection and Log-scaled mutation Genetic Algorithm (ESALOGA) that as an optimization package solves combinatorial problems. Section 5 presents simulated results in comparison to other variants of GA and three real coded optimizers concerning multi-modal benchmark functions. Finally, Sections 6 and 7 respectively conclude the paper and give future research directions and improvements.

## 2 Binary coded GA

A step by step operation of the binary coded GA is presented [9] that firstly allows to understand the concept of GA and symbiotic integration of different operators such as selection, crossover and mutation operators.

**Step 1:** At first, the parameters of GA are initialized as the crossover and mutation probabilities  $P_c$  and  $P_m$ , such that  $P_m \ll P_c$ , chromosomes in the population  $s$ , number of bits  $l$  to represent one variables as to decide the length of chromosomes, which is  $nl$  for  $n$  variables in the chosen problem. Termination criteria as the maximum number of generations that GA could proceed is selected based on the problem size.

**Step 2:** To start the evolution process, the fitness of each chromosome is calculated in the population. In this process, a part of binary chromosome representing a variable is decoded to express in decimal represented as  $d_n = \sum_{i=0}^{l-1} 2^i b_i^n$  where  $b_i^n \in \{0, 1\}$  belongs to the  $n^{th}$  variable. Values of  $n^{th}$  variable are obtained as the bound  $x_n^{(L)} \leq x_n \leq x_n^{(U)}$ , where  $n^{th}$  variable  $x_n$  is calculated as  $x_n = x_n^{(L)} + \frac{x_n^{(U)} - x_n^{(L)}}{2^n - 1} d_n$  based on its respective lower and upper bounds  $x_n^{(L)}$  and  $x_n^{(U)}$ . After converting the variable in a required domain, the associated objective function  $f(x)$  is calculated for all individuals represented by the chromosome strings in the population. For the minimization problem, the fitness function  $F_s$ , associated with  $s$  chromosome, is adopted as  $F_s = \frac{1}{1+f_s(x)}$ , which is the function of objective function  $f_s(x)$ .

**Step 3:** At this point, a selection operator selects the fittest chromosomes as the candidates go for mating, based on Roulette wheel selection [9, 30]. This is the first stage of GA process, where multiple different operators have been proposed, i.e., roulette wheel as in standard GA, tournament and uniform selection as a variant of GA. In the proposal, we introduce an enhanced selection scheme which is utilized after Step-4 instead in Step-3.

**Step 4:** In sequel, the crossover operator gives a number of strings from the mating pool using fixed crossover probability  $P_c$ . For the selected pair of candidates, knows as parents, a cross-site is generated randomly in the interval  $(0, nl - 1)$  and swapped the selected regions between two pairs. At this step, different crossover mechanisms have been proposed such as single-point, multi-point, uniform crossover, and etc.

The use of different crossover techniques makes the standard GA to enhance as the variant of it.

**Step 5:** After above serial processes, children chromosome strings arise as the result and the population of which is known as an intermediate population as taken in [9]. At this step, we have a pool of parents and resulting their offspring. Our proposal aims to answer which candidates should go to the next evolution phase as better parents.

**Step 6:** At this juncture, bitwise mutation is carried out, where as a result of mutation operator, a selected bit in the chromosome is flipped to opposite binary value based on a relatively low fixed mutation probability  $p_m$ . To make the process adaptive, based on the current status of the population, we propose a log scaled-mutation technique.

**Step 7:** Until the termination criterion is not reached, return to Step 2.

### 3 Selection and mutation schemes

In this section, we provide a step-by-step working principle of the proposed enhance selection and log-scaled mutation operators in order for providing an improved GA (ESALOGA) as a better optimization technique.

#### 3.1 Proposed selection operator

Based on the Metropolis algorithm [31], we focus on possible improvements of the GA for finding the optimal solution in due course of the cross-over while selecting chromosome strings. This keeps intact a high degree of diversity in selecting the children which are the most suitable when the chosen parents undergo a cross-over. To choose appropriate candidates from the current pool of parents and offspring a block diagrammed of the proposed selection strategy is given in Fig- 2. Mathematically, this is realized by introducing a selection probability as the Boltzmann probability distribution. Precisely, let  $T$  be the temperature, then the selection probability  $p(T)$  reads as the Maxwellian distribution

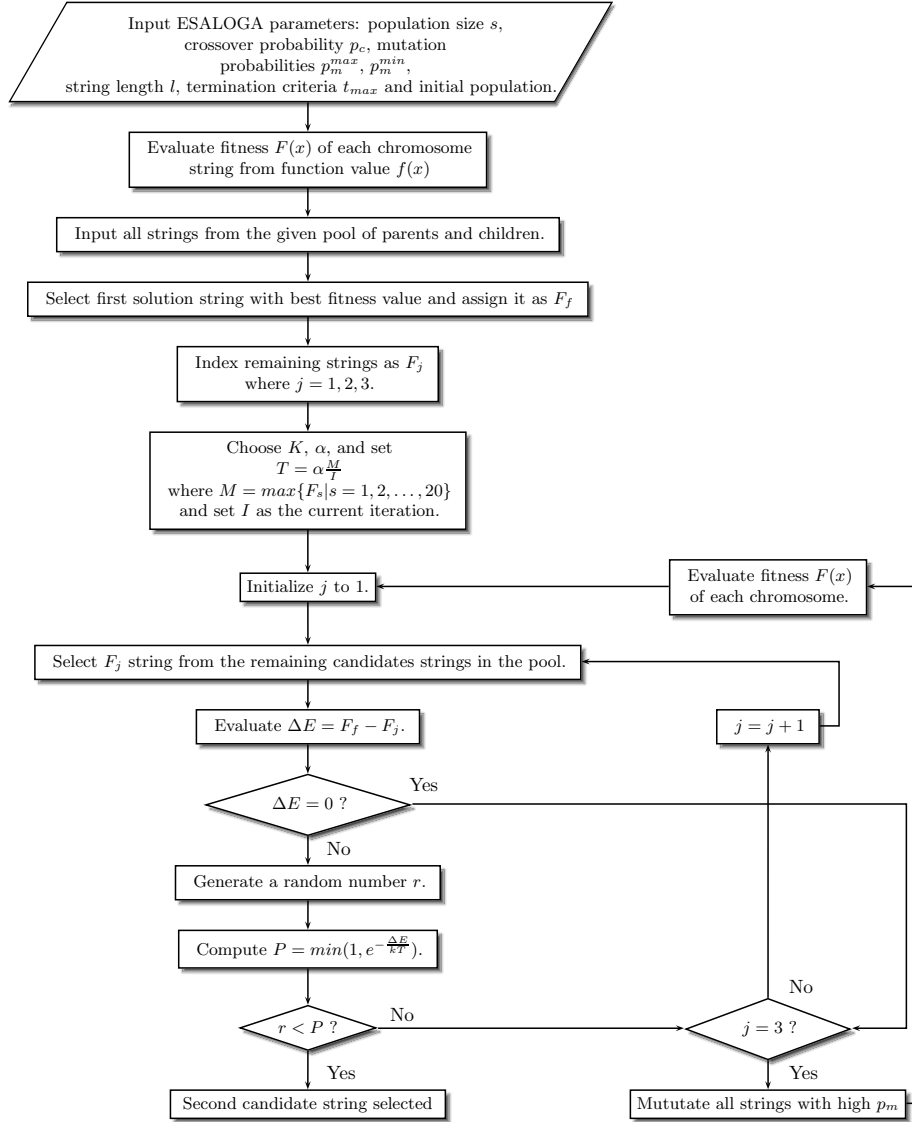
$$p(T) = e^{-\Delta E/kT}, \quad (1)$$

where  $\Delta E$  represents change in energy between the chosen parents and children. With the above probability  $p(T)$ , a set of selected strings passed to the next stage of evolution. It is worth mentioning that the principle of elitism [9] offers the best fitness value to the string for a given pool of parents and children. Following the Eqn. 1, the subsequent strings are selected that are the fittest string in the previous stage of the evolution. The proposed model is realized as per the following steps:

**Step 1:** Choose an initial value of the temperature  $T$  as

$$T = \alpha \frac{M}{I}, \quad (2)$$

where  $M$  is the maximum value of the fitness function  $\{F_s | s = 1, 2, \dots, 20\}$ ,  $I$  is the number of iterations and  $s$  labels the strings pertaining to the cross-over of a given population. Note that the initial value of the temperature  $T$  is taken as large as possible such that it decreases in the subsequent iteration to its desired value. Here, the proportionality constant  $\alpha$  is set as per the chosen algorithm.



**Fig. 1.** Flow diagram for selection strategy after crossover

**Step 2:** In order to find energy difference, one chooses  $j^{th}$  string in a given pool of parents and children and subtract the corresponding fitness value  $F_j$  of the  $j^{th}$  string to a priorly selected string  $F_f$ . In other words, the energy difference that governs the probability distribution is given by

$$\Delta E = F_f - F_j \quad (3)$$

with  $j = 1, 2, 3$ .

**Step 3:** Compute  $p$  as per the equation 1 and obtain its minimum value as

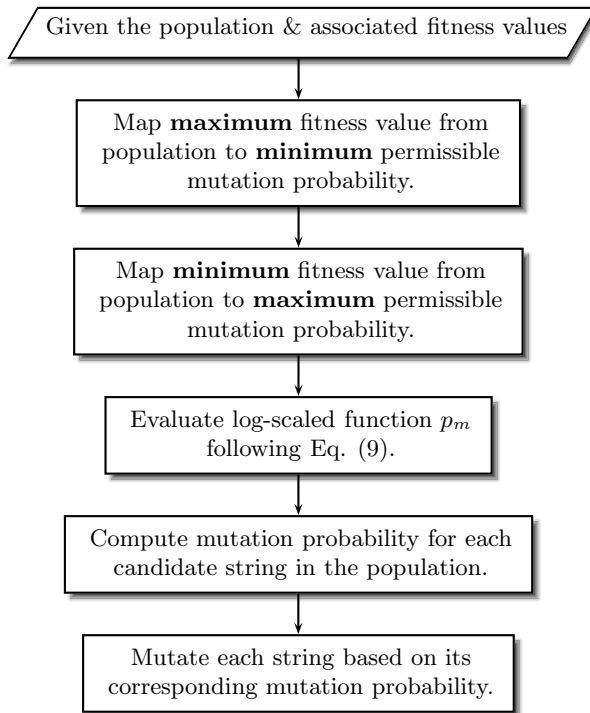
$$p = \min(1, e^{-\frac{\Delta E}{kT}}) \quad (4)$$

**Step 4:** Acquire a random number  $r \in (0, 1)$ .

**Step 5:** If a candidate string is selected, and the corresponding previously selected partner string is the fittest one, that is  $r < p$ .

**Step 6:** Else, go to Step 2, and repeat the search. In the case when none of the strings are selected, one increases the value of mutation probability  $p_m$ . In practical situations, we may consider the corresponding value  $p_m = 0.1$ .

**Step 7:** Finally, one selects the partner string chromosome by repeating Steps 2 to 6.



**Fig. 2.** Log-scaled mutation strategy

### 3.2 Proposed mutation operator

In this subsection, we offer log-scaled mutation strategy with the corresponding operations as below:

**Step 1:** Obtain the mutation probability for a given fitness value  $f_s(x)$  as per the transformation  $y_s = \log_{10} F_s$ .

**Step 2:** For the maximum value of the fitness  $F_s^{\max}$ , define  $y_s^{\max} = \log_{10} F_s^{\max}$ .

**Step 3:** Corresponding to the minimum fitness value  $F_s^{\min}$ , define  $y_s^{\min} = \log_{10} F_s^{\min}$ .

**Step 4:**  $y_s^{\max}$  is mapped to the minimum mutation probability  $p_m^{\min}$  such that the best candidates remain intact.

**Step 5:**  $y_s^{\min}$  is mapped to the maximum mutation probability  $p_m^{\max}$  such that the worst candidate mutate.

**Step 6:** Define a linear relationship between  $y_s$  and  $p_{m,s}$  as

$$p_{m,s} = \frac{p_m^{\max} - p_m^{\min}}{y_s^{\max} - y_s^{\min}} (y_s - y_s^{\min}), \quad (5)$$

where the ratio of  $p_m^{\max} - p_m^{\min}$  and  $y_s^{\max} - y_s^{\min}$  gives the  $\beta$  as the slop of the line plotted between  $p_{m,s}$  and  $y_s$ . This leads to the following linear equation

$$p_{m,s} = \beta y_s + \gamma, \quad (6)$$

where  $\gamma$  is the intercept of the line as in Eqn. 6 as

$$\gamma = -\frac{p_m^{\max} - p_m^{\min}}{y_s^{\max} - y_s^{\min}} y_s^{\min} \quad (7)$$

With the above slop  $\beta$  and intercept  $\gamma$ , the mutation probability  $p_{m,s}$  is obtained by the following logarithmic relation

$$p_{m,s} = \beta \log_{10} F_s + \gamma, \quad (8)$$

where  $s$  labels the undermining chromosome. Physically, this shows the inverse relation [9] between the fitness value  $F_s$  and the mutation probability  $p_{m,s}$ .

**Step 7:** This assigns a unique mutation probability  $p_{m,s}$  to each candidate strings in the range  $(p_m^{\min}, p_m^{\max})$ , viz. we have

$$p_m^{\min} \leq p_{m,s} \leq p_m^{\max} \quad (9)$$

**Step 8:** Finally, a diversity in the selected population is realized by a bitwise mutation process.

The fitness values of the strings are usually sparse, thus we propose a log-scale mutation operator. In this approach, we find that all mutation probabilities are kept in a specified range, irrespective of variations in the fitness values. This makes our proposal adaptive and yields an evolution from a premature to mature phase of a given population. In a nutshell, we have illustrated that there is a non-linear relationship between mutation probability and fitness value as far as evolutionary algorithms are concerned. In addition, it follows that the higher fitness value leads to the lower mutation probability. This indicates comparatively a larger search space while finding the global optimal solution.



## 4 Proposed GA (ESALOGA)

Based on the proposed enhanced selection and log-scaled mutation strategies, we provide below pseudo-code of the algorithm.

---

### Algorithm 1 Pseudo-code for the proposed ESALOGA

---

**Require:**  $N$ : the number of chromosomes,  $p_c$ : crossover probability,  $t_{max}$ : maximum iterations,  $p_m$ : mutation probability,  $p_m^{min}$ : lower bound on  $p_m$ ,  $p_m^{max}$ : upper bound on  $p_m$ ,  $b$ : number of bits to represent one variable,  $v$ : number of variables.

$P \leftarrow \text{round}(\text{rand}(N, b \cdot v))$ : initialize binary population randomly

- 1:  $GP \leftarrow \text{best of } [P]$ : GP belongs to the best solution in current  $P$
- 2: **for**  $i \leftarrow 1$  to  $t_{max}$  **do**
- 3:  $n \leftarrow 1$
- 4:   **while**  $n \leq N$  **do**
- 5:    $[\text{Parent1}, \text{Parent2}] \leftarrow \text{Selection}(P)$ : RW or Tournament selection operation
- 6:    $[\text{Children1}, \text{Children2}] \leftarrow \text{Xover}(\text{Parent1}, \text{Parent2})$ : Crossover operation
- 7:    $[\text{string1}, \text{string2}] \leftarrow \text{EnSelection}(\text{Parent1}, \text{Parent2}, \text{Children1}, \text{Children2})$ : Enhanced Selection operation to select two appropriate strings
- 8:    $P(n) \leftarrow \text{string1}$
- 9:    $P(n+1) \leftarrow \text{string2}$
- 10:    $n \leftarrow n+2$
- 11:   **end while**
- 12:  $P_m \leftarrow \text{LSMut}(P)$ : Log-scale mutation after crossover
- 13:  $P \leftarrow N$  best chromosomes of  $[P, P_m]$
- 14:  $GP(i) \leftarrow \text{best of } [P]$
- 15:   **if**  $\text{Fitness}(GP(i)) < \text{Fitness}(GP(i-1))$  **then**
- 16:      $GP(i) \leftarrow GP(i-1)$
- 17:   **end if**
- 18: **end for**
- 19: **return** GP

---

For a given input parameters randomly generates a binary initial population  $P$  with the fact that the candidate chromosome strings and mutation probability  $p_m$  is adaptively selected by enhanced selection operation (EnSelection) and given mutation range  $(p_m^{min}, p_m^{max})$ , respectively. Produce the mating pool for breeding for a given crossover probability  $p_c$ . Extract two parents from the mating pool using standard roulette wheel (RW) selection operator. Indeed, other selection schemes such as tournament and uniform could be adopted, as well, for better performance. Perform the single-point crossover operation to produce two children. Infact instead of sigle-point the use of two-points or uniform crossover operation may enhance the computational capability. At this junction, form a pool of two parents and produced their children choose two appropriate candidate strings using the enhanced selection (EnSelection) operator with probability  $p(T)$  as in Eqn. 1. As a result of this two appropriate candidates are selected to go in next evolution. When no chromosomes are selected from the pool, mutate all the strings with an increased mutation probability  $p_m$  and repeat the EnSelection operation. After this operation we get the intermediate population which is subjected to the mutation

operator with mutation probability  $p_m$ . Following the log-scaled strategy (LSMut), produce population of mutated string ( $P_m$ ), as in Algorithm 1. Taking best chromosome from the above two populations as shown in algorithm in line number (11). Repeat the steps until the termination criterion is reached.

## 5 Results and Discussion

In this section, we provide effectiveness of the proposed GA for various benchmark functions [9, 33]. Hereby comparing few variants of the GA, where they are distinguished based on their different selection and crossover strategies, an outline is given below: All the above variants are discussed in [32,33] and tested on the benchmark

**Table 1.** My caption

GA variants	SGA	VGA-1	VGA-2	VGA-3	VGA-4
Selection	RW	Random	RW	Random	Tournament
Crossover	Single-point	Two-points	Uniform	Uniform	Uniform

**Table 2.** Benchmark functions for testing ESALOGA

Functions	Mathematical Description
Himmelblau:	$f_1(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$ with variables limit $-6 \leq x_1, x_2 \leq 6$
Rastrigin:	$f(x_i) = An + \sum_{i=1}^n (x_i^2 - A \cos(2\pi x_i))$ with variables limit $-5.12 \leq x_1, x_2 \leq 5.12$
Ackley:	$f(x_1, x_2) = -20 \exp(-0.2 \sqrt{0.5(x_1^2 + x_2^2)}) - \exp(0.5(\cos(2\pi x_1) + \cos(2\pi x_2))) + e + 20$ $a = 20, b = 0.2, c = 2\pi$ , with variables limit $-35 \leq x_i \leq 35$
Beale:	$f(x_1, x_2) = (1.5 - x_1 + x_1 x_2)^2 + (2.25 - x_1 + x_1 x_2^2)^2 + (2.625 - x_1 + x_1 x_2^3)^2$ with variables limit $-4.5 \leq x_1, x_2 \leq 4.5$
Levi:	$f(x_1, x_2) = \sin^2(3\pi x_1) + (x_1 - 1)^2(1 + \sin^2(3\pi x_2)) + (x_2 - 1)^2(1 + \sin^2(2\pi x_2))$ with variables limit $-10 \leq x_1, x_2 \leq 10$
Goldstein:	$f(x_1, x_2) = (1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1 x_2 + 3x_2^2))(30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1 x_2 + 27x_2^2))$ with variables limits $-2 \leq x_1, x_2 \leq 2$
Styblinski -Tang:	$f(x) = \frac{1}{2} \sum_{i=1}^n (x_i^4 - 16x_i^2 + 5x_i)$ , with variables limit $-5 \leq x_i \leq 5$
Michalewicz:	$f(x) = -\sum_{i=1}^n \sin(x_i) \sin^{2m}\left(\frac{ix_i^2}{\pi}\right)$ with variables limit $0 \leq x_i \leq \pi$
Schaffer No2.:	$f(x) = 0.5 + \sum_{i=1}^{n-1} \frac{\sin^2(x_i^2 - x_{i+1}^2) - 0.5}{(1 + 0.001(x_i^2 + x_{i+1}^2))^2}$ with variables limit $-100 \leq x_i \leq 100$
Deceptive:	$f(x) = -\left[\frac{1}{n} \sum_{i=1}^n g_i(x_i)\right]^\beta$ with variables limit $0 \leq x_i \leq 1$ , and $\beta = 2$ ,
Keane Bump:	$f(x) = -\left \frac{\sum_{i=1}^n \cos^4(x_i) - 2 \prod_{i=1}^n \cos^2(x_i)}{(\sum_{i=1}^n ix_i^2)^{0.5}}\right $ subject to: $g_1(x) = 0.75 - \prod_{i=1}^n x_i < 0$ , $g_2(x) = \sum_{i=1}^n x_i - 7.5n < 0$

functions which are concisely tabulated in Table- 2. We firstly present the results on Goldsteinprice, Levi, Beale, Himmelblau, Ackley, and Rastrigin benchmark functions. Note that Rastrigin and Himmelblau functions are multimodal in their nature while the Ackly function possesses a large hole at its center with multi modularity. On the other hand, Beale function is a unimodular with four sharp peaks at the corners. Similarly, Levi function has a non-linear search space that may show a premature convergence in due course of the execution of our optimization algorithm. Equally, it is worth noticing that an optimization algorithm may get trapped in some of local minima of the objective function, which our proposal overcome by having a larger diversity as shown in Fig. 3 for different problems. Simulation results for comparative analysis of the ESALOGA with respect to standard GA, VGA-1, VGA-2, VGA-3, and VGA-4 is given in Table- 3 for 100 runs on aforementioned two variables problems.

Results are compared on Six attributes, i.e., the best achieved by algorithms, mean of the all solutions in 100 runs, standard deviation (Std) of the solutions achieved in 100 runs, reliability of the algorithms stand for the solution achieved by all lower than the mean of proposed GA, the worst achieved and at the last average time taken by all algorithms for 1000 evolution epochs. This follows from the average measurement techniques, giving a consistent and accurate determination of the approximate global optimal point as the effective of our proposed algorithm. Interestingly, while the SGA and other variants get trapped in one of their local optima, our proposed algorithm successfully terminates by locating the global optimum for various benchmark functions.

The corresponding comparative results of the diversity preservation is depicted in Fig. 3. In this figure, one can observe the spread of search for Himmelblau, Beale, Ackley and Levi functions. As we can see that SGA trapped at one point where ESALOGA examines different points for the global solution. Approximately, similar effect can be seen for other functions. We address the issue of premature convergence of the algorithm where the diversity preservation, where most of the GA variants behave similarly. Thus, we have proposed an enhance selection scheme to overcome this condition of premature convergence. We can equally maintain the diversity preservation adequately as shown in Fig. 3, which makes our algorithm relatively efficient. Further, we see from Fig. 3 that our proposal reveals various local and global optimal points of the aforementioned benchmark functions and offers a great diversity in searching process instead of getting the same point under different evolutions.

This yields an appropriate optimization with high diversity preservation in a given mating pool. We find an improved reliability (in percentage) as shown in Table-reftable2 in contrast to the standard GA and its other variants that get trapped in an intermediate suboptimal state at most of the time. Hereby, we find that the average performance of ESALOGA is comparable with the standard deviation. Also the time taken by ESALOGA is reasonable. Moreover, from the results on Himmelblau function, one can observe that ESALOGA tries the best to find a better solution, but on the cost of its runtime. It reveals that ESALOGA guarantees a better solution every time. As a mater of the fact, our algorithm yields an intelligent mechanism to come out from a suboptimal trap and local optima of a class of benchmark functions. By tuning the selective pressure to its higher value, we can generate a desired diversity in the population and

**Table 3.** Comparative simulation results of the proposed GA and other GA variants in 100 runs

	PGA	SGA	VGA-1	VGA-2	VGA-3	VGA-4
Goldsteinprice						
Best	3.0010	3.0010	3.0010	3.0010	3.0010	3.0010
Mean	3.0806	11.4169	11.5692	6.1199	5.7327	12.7232
Std	0.0734	17.1370	17.9966	14.0022	8.2900	17.3742
Reliability	60%	56%	56%	60%	60%	50%
Worst	3.313	88.868	84.080	89.541	32.634	76.699
time	1.2953	2.9910	3.2558	2.9816	2.9116	3.6707
Levi						
Best	7.8091e-04	5.5598e-05	5.5598e-05	5.5598e-05	5.5598e-05	5.5598e-05
Mean	0.0268	0.0555	0.0712	0.1220	0.1019	0.1432
Std	0.0270	0.1362	0.1733	0.2254	0.1790	0.3829
Reliability	60%	60%	70%	64%	54%	48%
Worst	0.110	0.725	0.725	0.725	0.725	2.600
time	1.3567	3.1313	3.7612	3.0177	3.3239	3.9555
Beale						
Best	3.1186e-05	8.0472e-05	8.0472e-05	3.1186e-05	3.1186e-05	2.1385e-04
Mean	0.0024	0.2249	0.2651	0.2645	0.1736	0.2841
Std	0.0027	0.3054	0.4824	0.3083	0.2815	0.3226
Reliability	60%	14%	14%	14%	24%	60%
Worst	0.012	0.926	2.689	0.816	0.926	0.974
time	1.3624	2.9880	3.2909	2.9839	2.9306	3.6881
Himmelblau						
Best	3.9863e-05	4.9682e-04	4.9682e-04	4.9682e-04	4.9682e-04	4.9682e-04
Mean	0.0633	0.2900	0.1968	0.2373	0.1486	0.6404
Std	0.2850	0.7672	0.5683	0.6091	0.4067	1.2703
Reliability	96%	76%	84%	76%	86%	64%
Worst	1.444	4.705	2.755	3.717	1.643	6.625
time	50.8745	3.6972	4.1740	3.4376	3.4056	4.5079
Ackley						
Best	0.0182	0.1982	0.1982	0.1982	0.1982	0.1982
Mean	0.0372	0.6684	0.4341	0.6788	0.3161	0.9028
Std	0.0097	1.1057	0.8200	1.0503	0.5920	1.2887
Reliability	42 %	0%	0%	0%	0%	0%
Worst	0.061	3.639	3.639	3.639	3.639	3.639
time	1.4144	3.0694	3.3770	3.3646	3.2705	3.6703
Rastrigin						
Best	0.0099	0.0104	0.0104	0.0104	0.0104	0.0104
Mean	0.2907	1.0351	1.2141	0.9595	1.5399	2.2707
Std	0.3054	1.0248	1.5045	1.0608	1.3773	2.1881
Reliability	66%	32%	32%	34%	20%	16%
Worst	1.0160	4.1020	7.9655	4.9817	5.0958	9.1854
time	1.3768	2.8542	3.1760	2.8413	3.0166	3.9116

scan entire search space while searching the global optimum. This provides an appropriate trade-off between the selective pressure and diversity pressure.

### 5.1 Comparison of proposed GA with other optimizers

In this section, we extend our algorithm to higher dimensions and provide comparison of ESALOGA with other optimizers involving certain complex functions, i.e., Rastigrin, Ackley, Schaffer no2 [34], Michalewicz [35], Styblinski-Tang [36], Deceptive [37], and constrained Keane's bump [38], as shown in Table -2. These functions have ability to extend in arbitrary dimensions with nonlinear analytical investigations. These functions are related to the real-world problems, for example, the Ackley function is considered as the free energy hypersurface of proteins. Most of the above test functions add the difficulty of being less symmetric and possess higher harmonics, which makes the functions difficult to solve and keep the environment uncertain. Namely, the Schaffer function has concentric barriers, whereby it capable to discriminate different optimizers. Hereby, we have tested our algorithm on highly nonlinear, multi-modal functions with a large number of local extrema. As mentioned above, one of them is the Michalewicz function, which is observed as a strange mathematical function having  $n!$  number of local optima in  $n$  dimensions. Our optimization algorithm has given an improved solution, as shown below in Table -4. Description of the Styblinski-Tang Function [36] is considered further.

Another complex function is the deceptive function, which finds its importance in discriminating different optimizers. As in Refs. [37, 38], it can be seen in the existing literature about its computational difficulties. Here, we have shown the results on the above complex functions, which qualify our algorithm as an apt global optimizer. In the sequel, we focus on the constrained complex function in multi-dimension. Namely, the Keane's bump function is considered as the test function. It is highly nonlinear and difficult to solve by the existing optimizers because its solution exists at a nonlinear boundary. Performance of the ESALOGA has been analyzed in comparison to VGA-4, probabilistic bee optimization (PBA) [40], invasive weed optimization (IWO) [41], and shuffle frog leap algorithm (SFLA) [42]. The comparative results of the above optimizers on ten dimensional test functions are shown in Table- 4. The parameters setting of all the optimizers are taken as per the followings:

1. VGA-4 parameters:
  - (a) Fifty chromosomes are taken in a population
  - (b) Crossover probability is fixed at 0.8 to form the mating pool
  - (c) Mutation probability is taken as 0.02
2. PBA parameters:
  - (a) Number of scout bees are 50
  - (b) Recruited bees scale are defined as  $\text{round}(0.3 \times 50)$
  - (c) Neighborhood radius is set as  $0.1 \times (\text{maximum variable value} - \text{minimum variable value})$
  - (d) Neighborhood radius damp rate is 0.9
3. IWO parameters:
  - (a) Population size is taken as 50
  - (b) Minimum and maximum numbers of the seeds are 0 and 5 respectively
  - (c) Variance reduction exponent is set to 2
  - (d) Initial and final values of the standard deviation are 0.5 and 0.01 respectively
4. SFLA parameters:

- (a) Memplex size is 25
  - (b) Number of memplexes is 2
  - (c) Number of parents are defined as the maximum of rounded value of  $(0.3 \times 25)$  or 2
  - (d) Number of off-springs is taken as 3
  - (e) Maximum number of iterations is 5
5. proposed ESALOGA parameters:
- (a) 50 chromosomes are taken in the population
  - (b) Crossover probability is 0.8 to form a mating pool
  - (c) Mutation probability is adaptively defined between 0 to 0.05 by our proposed mutation scheme
  - (d) Mutation probability during enhance selection procedure is 0.02.

In one run of the optimization, all optimizers give the solution in five hundred generations. We run the proposed algorithm for all the above mentioned benchmark functions for fifty times to see the performance statistics. Hereby, we compare the results on all the selected benchmark functions. Through the observation of the Table -4, we can deduce the preeminence of the ESALOGA over PBA, IWO, SFLA and GA for the above class of test functions. Comparison made on the five indices named as Best, Worst, Mean, which is achieved in 50 runs of the optimizer, Std the standard deviation of solutions in 50 runs by optimizer, and Consistency which is defined as how many times the optimizer is qualified as an expected solution (in percentage).

Based on the observations as in Table- 4, we can extract the following comparative results:

1. ESALOGA is highly consistent than the other optimizers. In comparison other optimizers, we have observed with the presented results that the ESALOGA performs well for multimodal functions, which are highly complex functions in their nature according to the literature [37, 38].
2. For Styblinski-Tang, Ackley, Rastrigin and Deceptive functions, we find that no other optimizer that the VGA-4 gives acceptable results as in Table 4. Here, ESALOGA gives the optimal solution with a high consistency and low standard deviation. Table 4 shows that the consistency of ESALOGA is 90%, 90%, 99%, and 100% for Styblinski-Tang, Rastrigin Ackley and Deceptive functions respectively.
3. For Michalewicz function, the best optimization is given by ESALOGA with a consistent mean around -9.0075, which is the best in comparison to all other optimizers.
4. Schaffer function No. 2 is another highly complex function, which we have solved by the ESALOGA with better results than the other above mentioned optimizers.
5. On the highly complex constraint test function named as Kean-bump function, the ESALOGA gives outstanding results over other optimizers. Note that only GA tries to compete with the results of the ESALOGA.
6. Overall, statistical results of ESALOGA are far better than the other optimizers, as well.

**Table 4.** Comparative results on ten variables for fifty runs

	VGA-4	PBA	IWO	SFLA	PGA
<b>Styblinski-Tang Function</b>					
Best	-389.2077	-261.6483	-377.5249	-377.5249	-391.6528
Worst	-374.5288	-176.9046	-320.9780	-374.5288	-376.9688
Mean	-383.0037	-218.7373	-352.0788	-354.9062	-385.3267
Std	4.2495	22.0725	16.9144	15.4860	3.4056
Consistency (Solution <-383)	55%	0%	0%	0%	90%
<b>Michalewicz Extension function</b>					
Best	-9.5033	-3.4877	-9.3631	-9.2164	-9.6575
Worst	-8.1878	-2.2156	-7.9995	-8.1878	-8.2459
Mean	-8.9632	-2.8983	-8.8179	-8.6147	-9.0075
Std	0.3990	0.3521	0.4090	0.4242	0.3343
Consistency (Solution <-9)	55%	0%	40%	30%	65%
<b>Ackley function</b>					
Best	0.0016	2.3168	0.0020	0	1.335e-04
Worst	2.0225	19.7360	18.8521	1.6538	1.6538
Mean	0.1086	11.6177	12.7090	0.4520	0.0828
Std	0.4506	4.8105	8.5472	0.8391	0.3698
Consistency (Solution <0.1)	95%	0%	30%	75%	99%
<b>Rastrigin function</b>					
Best	3.0071	9.9496	0.9955	2.9849	1.0173
Worst	21.2696	34.8234	16.9149	21.2696	14.2134
Mean	10.8882	24.4759	8.7562	14.8746	6.3935
Std	5.0115	5.7601	3.7172	8.1695	3.5663
Consistency (Solution <10)	55%	50%	75%	35%	90%
<b>Schaffer function No. 2</b>					
Best	-3.9918	-1.0227	-1.1854	-3.4150	-3.7789
Worst	-2.1046	0.0065	-0.1801	-2.1046	-2.6381
Mean	-3.1594	-0.0941	-0.5848	-2.6446	-3.3691
Std	0.4458	0.2305	0.2734	0.5166	0.2948
Consistency (Solution <-3)	55%	0%	0%	25%	75%
<b>Deceptive function</b>					
Best	-0.9255	-0.4140	-0.7724	-0.8464	-0.9255
Worst	-0.7483	-0.2729	-0.7040	-0.7483	-0.7187
Mean	-0.8196	-0.3185	-0.7259	-0.7853	-0.7955
Std	0.0394	0.0389	0.0247	0.0326	0.0399
Consistency (Solution <0.8)	40%	0%	0%	10%	100%
<b>Keane Bump function</b>					
Best	-0.7257	-0.2368	-0.7492	-0.7038	-0.7405
Worst	-0.6290	-0.1238	-0.2740	-0.6014	-0.6014
Mean	-0.6818	-0.1750	-0.5778	-0.5532	-0.6856
Std	0.0292	0.0278	0.1532	0.1073	0.0357
Consistency (Solution <0.6)	30%	0%	25%	15%	55%

## 6 Conclusion

In this paper, we have given an improved search technique based on biological evolution. This is well suited to optimize multi-variable objective functions with and without discontinuities. As a matter of the fact, the proposed operators are flexible in finding the global minimum solution of a benchmark function. Hereby, our proposition gives an improved technique for solving optimization problems. Further, we have given simulation results of our proposal as a variant of the standard GA. As the verification of the same, we have enlisted the global solution of various of two variables bench-mark functions.

From the simulated results, it is found that our method precisely locates the optimal points of multi-modal bench-mark functions. Hereby, various drawbacks of the binary-coded GA including imprecision and inconsistency are taken care by Metropolis scheme. This provides an enhanced selection and adaptive log-scale mutation scheme. Subsequently, the global optimal solution is obtained with an acceptable value of selection pressure. In other words, our proposal is a meta-heuristic approach as far as the global optimization problems are concerned. Indeed, this gives an improved precision and consistency as revealed via the simulated results.

## 7 Future Scope

Proposed GA has a considerable scope of further improvement as discussed in this section. The first stage of improvement belongs to the parallel population approach which may give a better solution. To introduce more diversity, random selection, tournament selection can be tested instead of roulette wheel selection before crossover, where we are proposing selection after crossover. This is taken as complimentary selection scheme for introducing more diversity after crossover. In next paper, we will test the above specified selection strategies with proposed GA and compare with the different variants of the available GA. The second improvement is at the stage of crossover where different crossover techniques such as single-point, multi-point, uniform, mid-point techniques can be tested to see the superiority of proposed GA over other variants.

Results on the limited functions showing the proposed GA superiority over SGA and other variants of GA. Thus, the insertion of enhanced selection scheme as treated as complimentary selection after crossover, the log mutation scheme in the structure of other GA variants may give better results over others. Moreover, performance of the proposed GA can be increased by utilizing binary tree memory. Thus, we observe that the proposed GA has a wide scope of improvements and it may further emerge as a dominant optimization algorithm for large scale complex problems from sociology, engineering, Topology, Graphs, Biology, and etc. At this juncture, we anticipate that our proposal finds various applications in real world industrial problems such as power systems, its transmission expansion planning, data systems and wireless technology.

## References

1. Bill, N.M., David, M.R.: Total productive maintenance: A timely integration of production and maintenance. *Production and Inventory Management Journal*. 33(4), 6 – 10 (1992)



2. Bevilacqua, M., Braglia, M.: The analytic hierarchy process applied to maintenance strategy selection. *Reliability Engineering & System Safety*. 70(1), 71 – 83 (2000)
3. Doganay, K.: Applications of optimization methods in industrial maintenance scheduling and software testing. Mlardalen University Press Licentiate Theses, School of Innovation, Design and Engineering. 180 (2014)
4. Shen, M., Peng, M., Yuan, H.: Rough set attribute reduction based on genetic algorithm. *Advances in Information Technology and Industry Applications, the series Lecture Notes in Electrical Engineering*. 136, 127 – 132 (2012)
5. Sobh, T., Elleithy, K., Mahmood, A., Karim, M.: Innovative algorithms and techniques in automation, *Industrial Electronics and Telecommunications* (2007)
6. Hillier, M.S., Hillier, F.S.: Conventional optimization techniques, evolutionary optimization. *International Series in Operations Research & Management Science*. 48, 3 – 25 (2002)
7. Miettinen, K., Neittaanmaki, P., Makela, M.M., Periaux, J.: Evolutionary algorithms in engineering and computer science: recent advances in genetic algorithms, *Evolution Strategies, Evolutionary Programming, Genetic Programming and Industrial Applications*, John Wiley & Sons (1999)
8. Kar: Genetic algorithm application. <http://business-fundas.com/2011/genetic-algorithm-applications/> (2016). Accessed 27 June 2016
9. Deb, K.: Optimization for engineering design: algorithms and examples. Prentice hall of India, Private limited, New Delhi, (2005)
10. Tiwari, B.N.: Geometric perspective of entropy function: embedding, spectrum and convexity, LAP LAMBERT Academic Publishing, ISBN-13: 978-3845431789 (2011)
11. Gupta, N., Tiwari, B.N., Bellucci, S.: Intrinsic geometric analysis of the network reliability and voltage stability, *International Journal of Electrical Power & Energy Systems*. 44(1), 872 – 879 (2010)
12. Bellucci, S., Tiwari, B.N., Gupta, N.: Geometrical methods for power network analysis, *SpringerBriefs in Electrical and Computer Engineering*, ISBN: 978-3-642-33343-9 (2013)
13. Nelson, B.L.: Optimization via simulation over discrete decision variables, *Tutorials in Operation Research*, INFORMS, 193 – 207 (2010)
14. Gupta, N., Shekhar, R., Kalra, P. K.: Computationally efficient composite transmission expansion planning: A Pareto optimal approach for techno-economic solution. *Electrical Power and Energy Systems*. 63, 917 – 926 (2014)
15. Neeraj Gupta, Rajiv Shekhar, Prem K. Kalra, Congestion management based roulette wheel simulation for optimal capacity selection: probabilistic transmission expansion planning, *Electrical Power and Energy Systems*, vol. 43, 2012, 1259 – 1287.
16. Goldberg, D.E.: Genetic algorithms in search optimization and machine learning, Addison-Wesley, Reading, MA (1989b).
17. Chung, H.S.H., Zhong, W., Zhang, J.: A novel set-based particle swarm optimization method for discrete optimization problem. *IEEE Trans. on Evol. Computation*. 14(2), 278 – 300 (2010)
18. Liang, Y.C., Smith, A.E.: An ant colony optimization algorithm for the redundancy allocation problem (RAP). *IEEE Transactions on Reliability*. 53(3), 417 – 423 (2004)
19. Sharapov, R.R.: Genetic algorithms: basic ideas, variants and analysis, Source: Vision Systems: Segmentation and Pattern Recognition, ISBN 987-3-902613-05-9, Edited by: Goro Obinata and Ashish Dutta, pp.546, I-Tech, Vienna, Austria, June 2007. Open Access Database [www.i-techonline.com](http://www.i-techonline.com)
20. Holland, J.H.: Adaptation in natural and artificial systems, University of Michigan Press, Ann. Arbor, MI (1975)
21. Goldberg, D.E., Lingle, R.: Alleles, loci, and the TSP, in: Proc. 1st Int. Conf. on Genetic Algorithms. 154 – 159 (1985)
22. Malhotra, R., Singh, N., Singh, Y.: Genetic algorithms: concepts, design for optimization of process controllers. *Computer and Information Science*. 4(2), 39 – 54 (2011)

23. Spears W.M., De Jong, K.A.: On the virtues of parameterized uniform crossover, in: Proc. 4th Int. Conf. on Genetic Algorithms (1994)
24. Gupta, D., Ghafir, S.: An Overview of methods maintaining diversity in genetic algorithms. *Int. Journal of Emerging Technology and Advanced Engineering*. 2(5), 263 – 268 (2012)
25. Ming, L., Junhua, L.: Genetic algorithm with dual species. In *International Conference on Automation and Logistics Qingdao*. 2572 – 2575 (2008)
26. Cantu-Paz, E.: A survey of parallel genetic algorithms. *Calculateurs paralleles, reseaux et systems repartis*. 10(2), 141 – 171 (1998)
27. Aggarwal, S., Garg, R., Goswami, P.: A review paper on different encoding schemes used in genetic algorithms. *International journal of advanced research in computer science and software engineering*. 4(1), 596 – 600 (2014)
28. Baluja, S., Caruana, R.: Removing the genetic form the standard genetic algorithm, in Proc. 12th Int. Conf. on Machine Learning, 38 – 46 (1995)
29. Srinivas, M., Patnaik, M.: Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE transaction on systems, man and cybernetics*. 24(4), 656 – 667 (1994)
30. Goldberg, D.E., Sastry, K., Kendall, G.: Chapter 4 Genetic Algorithms, in Burke, E.K., Kendall, G. (eds.), *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*. Springer, Science + Business Media, NY (2014)
31. Cipra, B.A.: The Best of the 20th Century: Editors Name Top 10 Algorithms, *SIAM News*. <https://www.siam.org/pdf/news/637.pdf> Volume 33(4) 2016. Accessed 27 June 2016
32. Man, K.F., Tang, K.S., Kwong, S.: Genetic algorithm: concepts and applications, *IEEE Transactions on Industrial electronics*. 43(5), 519 – 534 (1996)
33. Jamil, M., Yang, X.: A Literature survey of benchmark functions for global optimization problems. *Int. Jour. of Math. Modelling and Numerical Optimization*. 4(2), 150 – 194 (2013)
34. <https://www.sfu.ca/ssurjano/schaffer2.html>
35. <https://www.sfu.ca/ssurjano/michal.html>
36. <https://www.sfu.ca/ssurjano/stybtang.html>
37. Iclnzan, D. (2014), Global optimization of multimodal deceptive functions. In: Blum C., Ochoa G. (eds) *Evolutionary Computation in Combinatorial Optimisation. EvoCOP 2014. Lecture Notes in Computer Science*, vol 8600. Springer, Berlin, Heidelberg
38. Li, Y. (2005), The deceptive degree of the objective function. In: Wright A.H., Vose M.D., De Jong K.A., Schmitt L.M. (eds) *Foundations of Genetic Algorithms. FOGA 2005. Lecture Notes in Computer Science*, vol 3469. Springer, Berlin, Heidelberg
39. Mishra, S.K. (2007). "Minimization of Keane's bump function by the repulsive particle swarm and the differential evolution methods", May 2007. SSRN: <http://ssrn.com/abstract=983836>.
40. Karaboga, D. and Akay, B. (2009). A comparative study of artificial bee colony algorithm, *Applied Mathematics and Computation*, Volume 214, Issue 1, 1 August 2009, Pages 108-132.
41. BozorgHaddad, O., Solgi, M. and Loiciga, H.A. (2017). Invasive weed optimization. *Meta-Heuristic and Evolutionary Algorithms for Engineering Optimization*, pp.163-173, Wiley.
42. Eusuff, M., Lansey, K. and Pasha, F. (2006). Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization. *Engineering optimization* 38(2), pp.129-154. Taylor & Francis.

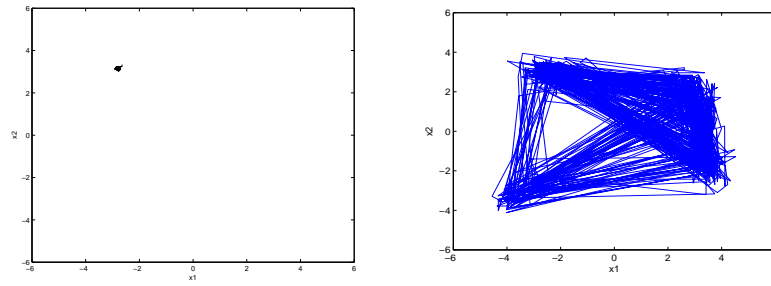


Fig. 1.1 Himmelblau function

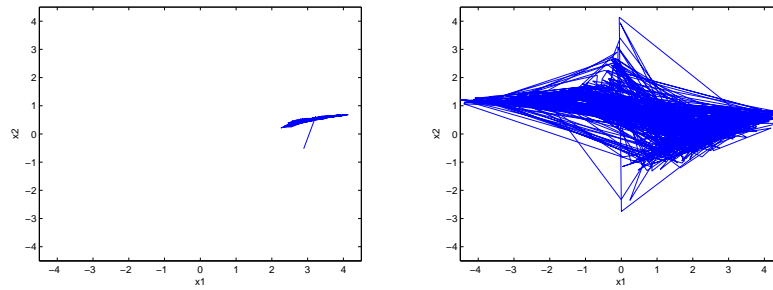


Fig. 1.2 Beale function

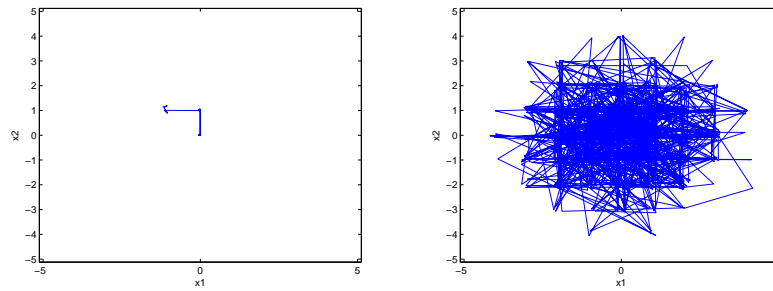


Fig. 1.3 Ackley function

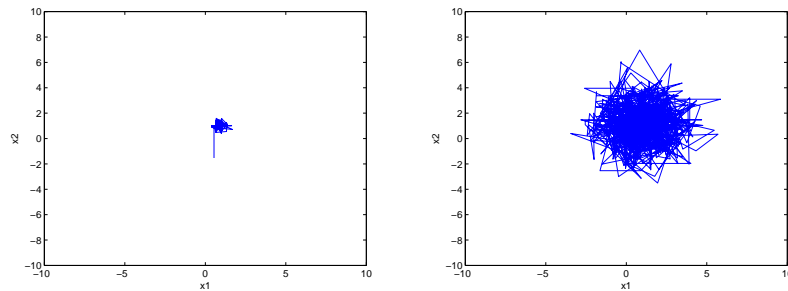


Fig. 1.4 Levi function

**Fig. 3.** Comparative result for the diversity preservation for the same number of generations (Left: Standard GA, Right: Proposed GA)