Mendelian Evolutionary Theory Optimization Algorithm

Neeraj Gupta [1], Mahdi Khosravy [2], Nilesh Patel [1], Nilanjan Dey [1], and Om Prakash Mahela [1]

[1]Affiliation not available
[2]Osaka University

October 30, 2023

Abstract

This study presented a new multi-species binary coded algorithm, Mendelian Evolutionary Theory Optimization (METO), inspired by the plant genetics. This framework mainly consists of three concepts: First, the "denaturation" of DNA's of two different species to produce the hybrid "offspring DNA". Second , the Mendelian evolutionary theory of genetic inheritance, which explains how the dominant and recessive traits appear in two successive generations. Third, the Epimuation, through which organism resist for natural mutation. The above concepts are reconfigured in order to design the binary meta-heuristic evolutionary search technique. Based on this framework, four evolutionary operators – 1) Flipper, 2) Pollination, 3) Breeding, and 4) Epimutation – are created in the binary domain. In this paper, METO is compared with well-known evolutionary and swarm optimizers 1) Binary Hybrid GA (BHGA), 2) Bio-geography Based Optimization (BBO), 3) Invasive Weed Optimization (IWO), 4) Shuffled Frog Leap Algorithm (SFLA), 5) Teaching-Learning Based Optimization (TLBO), 6) Cuckoo Search (CS), 7) Bat Algorithm (BA), 8) Gravitational Search Algorithm (GSA), 9) Covariance Matrix Adaptation Evolution Strategy(CMAES), 10) Differential Evolution (DE), 11) Firefly Algorithm (FA) and 12) Social Learning PSO (SLPSO). This comparison is evaluated on 30 and 100 variables benchmark test functions, including noisy, rotated, and hybrid composite functions. Kruskal Wallis statistical rank-based non-parametric H-test is utilized to determine the statistically significant differences between the output distributions of the optimizer, which are the result of the 100 independent runs. The statistical analysis shows that METO is a significantly better algorithm for complex and multi-modal problems with many local extremes.

Noname manuscript No.
(will be inserted by the editor)

Mendelian Evolutionary Theory Optimization Algorithm⋆

Neeraj Gupta · Mahdi Khosravy* · Nilesh Patel · Nilanjan Dey · Om Prakash Mahela

Received: date / Accepted: date

Abstract This study presented a new multi-species binary coded algorithm, *Mendelian Evolutionary Theory Optimization (METO)*, inspired by the plant genetics. This framework mainly consists of three concepts: *First*, the "denaturation" of DNA's of two different species to produce the hybrid "offspring DNA". *Second* , the Mendelian evolutionary theory of genetic inheritance, which explains how the dominant and recessive traits appear in two successive generations. *Third*, the *Epimuation*, through which organism resist for natural mutation. The above concepts are reconfigured in order to design the binary meta-heuristic evolutionary search technique. Based on this framework, four evolutionary operators – 1) Flipper, 2) Pollination, 3) Breeding, and 4) Epimutation – are created in the binary domain. In this paper, METO is compared with well-known evolutionary and swarm optimizers 1) Binary Hybrid GA (BHGA), 2) Bio-geography Based Optimization (BBO), 3) Invasive Weed Optimization (IWO), 4) Shuffled Frog Leap Algorithm (SFLA), 5) Teaching-Learning Based Optimization (TLBO), 6) Cuckoo Search (CS), 7) Bat Algorithm (BA), 8) Gravitational Search Algorithm (GSA), 9) Covariance Matrix Adaptation Evolution Strategy(CMAES), 10) Differential Evolution (DE), 11) Firefly Algorithm (FA) and 12) Social Learning PSO (SLPSO). This comparison is evaluated on 30 and 100 variables benchmark test functions, including noisy, rotated, and hybrid composite functions. Kruskal Wallis statistical rank-based non-parametric H-test is utilized to determine the statistically significant differences between the output distributions of the optimizer, which are the result of the 100 independent runs. The statistical analysis shows that METO is a significantly better algorithm for complex and multi-modal problems with many local extremes.

Keywords Mendelian evolutionary theory · rehabilitation · binary coded optimizer · pollination · meta-heuristic optimization · multi species · artificial DNA

1 Introduction

Optimization plays an essential role in achieving accuracy and increasing efficiency of systems. Under the classes of real and binary coded schemes, the literature proposes a variety of meta-heuristic population-based Evolutionary Algorithms (EAs) [1–3] i.e. Genetic Algorithm (GA) [4, 5], Memetic algorithm (MA) [6], PSO [7] etc. Although the population-based EAs have been being widely accepted by the researchers and industries from different fields [8]. Literature revels the limitations of meta-heuristic algorithms, where it is hard to solve multi modal functions [9–11]. These limitations invoke the researcher to find the better optimization technique, which should be able to offer better results. In this regard, this paper makes an effort to present a better optimization technique to handle multi-modularity of the objective functions.

Literature shows tremendous effort from nineties where we can observe many innovations in EAs and Swarm optimization techniques with their variants [12–18]. After ob-

⋆ Preprint submitted to Soft Computing journal, Springer

N. Gupta, N. Patel
School of Engineering and Computer Science, Oakland University, USA

* M. Khosravy (corresponding author)
Graduate School of Engineering, Osaka University, Japan
E-mail: mahdi.khosravy@nanase.comm.eng.osaka-u.ac.jp

N. Dey
Techno International New Town, Kolkata, India

O.P. Mahela
Power System Planning Division, Rajasthan Rajya Vidyut Prasaran Nigam Ltd., Jaipur 302005, India

Table 1: Table of Acronyms, Variables, and notations

AS	Anti-Sense Strand	BBO	Bio-geography Based Optimization
BHGA	Binary Hybrid Genetic Algorithm	CMAES	Covariance Matrix Adaptation Evolution Strategy
CS	Cuckoo Search	DE	Differential Evolution
DG	Dominant genes	DNA	Deoxyribonucleic acid
EA	Evolutionary Algorithm	FA	Firefly Algorithm
GSA	Gravitational Search Algorithm	IWO	Invasive Weed Optimization
MA	Memetic algorithm	METO	Mendelian Evolutionary
RG	Recessive genes		Theory Optimization
SFLA	Shuffled Frog Leap Algorithm	SIA	Swarm Intelligence Algorithm
SLPSO	Social Learning Particle Swarm Optimization	SS	Sense Strand
		TLBO	Teaching-Learning Based Optimization
α	maximum function evaluations	β	current function evaluation
δ	Flipping probability	$\mho$	Population of DNAs
$\mathscr{B}$	Cross-Breeding	$\mathbb{B}$	best solution
$\mathscr{C}$	consistency	$\tilde{\chi}^2$	chi-square value
d	Representing DNA number	$\mathscr{E}$	Epimutation
		$\mathscr{F}$	Flipper
$F0$	Current parents' population	$F1$	Next generation offspring
$F2$	Next to Next generation offspring	$\mathbb{G}$	Genotype domain
		g	Gene
H_0	Null hypothesis	$\mathscr{H}$	Heredity
l	Gene location (locus)	l_{start}	First bit of
l_{last}	last bit of	μ	mean
N_v	Number of DNA	N_b	DNA bits number
$\mathscr{O}$	Self-Breeding	p	p-value
$\mathscr{P}$	Pollination	ψ	distribution median
$\mathbb{R}$	Phenotype domain	r	Population of SS
$\mathscr{W}$	worst result	x	Real variable
$\hat{x}$	Binary vector to real number conversion	v	Population of AS

Sub and Super Script:

$F11$	$F1$ generation 1^{st} offspring
$F12$	$F1$ generation 2^{nd} offspring
i, j, k	species
l	Lower limit of something
u	Upper limit off something
n	Number of chromosomes in a population

serving the computational power of population based optimization algorithms in last two decades, variants of swarm optimization [7,16], Shuffled Frog Leaping Algorithm (SFLA) [19], Binary Hybrid GA (BHGA) [20], Biogeography-Based Optimization (BBO) [21], Cuckoo Search (CS) [22], Bat Algorithm (BA) [23], Teaching Learning Based Optimization (TLBO) [24], and etc. have been proposed, as in survey papers and recent books [3, 12, 25–28].

Recent papers [29–31] show the comparative evaluations of the recent meta-heuristic optimizer. To add the state-of-the-art, we inspired from the evolution theory of plant genetics based on Mendel's inheritance law to propose a genetically evolved optimization algorithm. In this algorithm, the evolution process takes place by interbreeding the plants of different species [32].To design a novel Evolutionary algorithm (EA), we redefine the biologically inspired metaphors in the binary domain to implement as computer program. Due to its binary structure, the proposed five operations: flipping, pollination, breeding, discriminating and Epimutation [33–36] have encoding and decoding techniques like the Genetic algorithm (GA) [20], but in a different working structure wherein the ancestors' memory is transferred in two consecutive generations, $F1$ and $F2$ offspring as METO produces two generations offspring in the sequence. This makes the METO belongs to a different class from the GA. METO is a gradient-free method that does not require the function differential, thus best suited for the discontinuous and multi-modal problems as well.

The evaluative study of METO performs an analytical comparison with twelve optimizers, including the state of the art techniques as described briefly in Table-2. For this several classes of benchmark test functions are considered including noisy, rotated, and hybrid composite functions [37–39]. The following distinctive features describe METO algorithm:

(i) It is a binarily coded optimizer and explores the transformed genome search space instead of real.

(ii) Mendel deduced his theory by the experimentation on the pea plants wherein genetic information exchange took place by breeding the plants of different species. Based on his experiments, METO employs multiple populations, where each population correspond to a particular species. This is the base of METO algorithm.

(iii) Instead of producing one generation offspring in an epoch through crossover/breeding, which is usual in GA and evolutionary optimizers, METO produces two consecutive generations offspring in an evolution epoch. Inspired from Mendel's experiments, first generation offspring, F1, is produced by cross-breeding of F0 generation parents, and the second generation offspring, F2, by self-breeding of the F1 generation parents.

(iv) An organism[1] in the population is represented by the complementary double strands artificial DNA.

(v) DNA of F1 generation offspring, as a result of cross-breading, is formed by combining the opposite strands of DNA of two parents [33]. Based on Mendel experiments, breeder parents belong to the two different

[1] In this paper, we interchangeably use the organism for the plant, where a plant is biologically represented by the chromosome.

Table 2: Optimizers for the comparison

Covariance Matrix Adaptation Evolutionary Strategy (CMAES) [48, 49]	Year: 2003	Type: EA	Single population
	Inspiration: Covariance matrix (CM) is used to define the pairwise dependencies between the variables. Here, adaption of this matrix belongs to CMA.		
	Parameter values: Number of parents(λ) = PopulationSize/2; Parent Weights (w) =(log(λ +0.5)-log(1: λ))/sum(log(λ +0.5)-log(1: λ));Number of Effective Solutions (eff)= 1/sum(w^2); Step Size Control Parameters (c_sigma and d_sigma): sigma0= 0.3$\times$ (x^U - x^L); cs=(eff+2)/(nVar+ eff + 5); ds=1+cs+2$\times$max(sqrt((eff-1)/(nVar + 1))-1,0); ENN=sqrt(nVar)$\times$(1-1/(4$\times$nVar)+1/(21$\times$nVar^2)); Here, nVar stands for number of Variables Covariance Update Parameters: cc=(4+ nP_eff/nVar)/(4+nVar+2$\times$eff/nVar); c1=2/((nVar+1.3)^2+eff); alpha_nP= 2; cmu=min(1-c1, α_λ $\times$(eff-2+1/eff)/((nVar+2)^2+α_λ $\times$ eff/2)); threshold for updating covariance matrix = (1.4+2/(nVar+1))$\times$ENN;		
DE with Dither technique (DED) [50]	Year: 2005	Type: EA	Single population
	Variant: DE/rand/1/bin, Parameters: weighting factor =interval $[0.5, 1.0]$, crossover constant $= 0.9$		
	Parameter values: Lower Bound of Scaling Factor $= 0.2$; Upper Bound of Scaling Factor $= 0.8$; Crossover Probability= 0.2;		
Invasive Weed colonization (IWO) [41]	Year: 2006	Type: EA	Single population
	Inspiration: Inspired by dynamic growth of weeds		
	Parameter values: Maximum population size $= (1+0.2)\times$ Population Size; Minimum Number of Seeds $= 0$; Maximum Number of Seeds= 5; Variance Reduction Exponent= 2; Initial Value of Standard Deviation = 0.5; Final Value of Standard Deviation $= 0.001$		
Shuffled Frog Leap Algorithm (SFLA) [19]	Year: 2006	Type: MA	Multiple population
	Inspiration: Frog foraging behavior with mimetic algorithm concept		
	Parameters: memeplex size, parents number, offspring number, step size, iteration number		
Biography Based Optimization (BBO) [21]	Year: 2008	Type: EA	Single population
	Inspiration: Biogeography; the study of the distribution of biological species through time and space		
	Parameter values: Keep rate = 0.2; Emmigration Rates = linspace(1,0,PopulationSize); Immigration Rates= 1- Emmigration Rates; alpha = 0.9; mutation probability = 0.1; sigma = 0.02$\times$(x^U - x^L);		
Cuckoo Search (CS) [22, 42]	Year: 2009	Type: SIA	Single population
	Inspiration: Obligate brood parasitism of some cuckoo species by laying their eggs in the nests of other host birds [2]		
	Parameter values:Discovery rate of alien eggs/solutions $= 0.25$; Number of nests = Population Size; Levy exponent and coefficient $\beta = 3/2$; $\sigma = (\gamma(1+\beta)\times\sin(\pi\times\beta/2)/(\gamma((1+\beta)/2)\times\beta\times 2^{(}(\beta-1)/2)))^{(}1/\beta)$;		
Firefly Algorithm (FA) [47]	Year: 2009	Type: SIA	Single population
	Inspiration: Flashing behavior of fireflies		
	Parameter values: Light Absorption Coefficient $= 1$; Attraction Coefficient Base Value $= 0.2$; Mutation Coefficient $= 0.2$; Mutation Coefficient Damping Ratio $= 0.98$; Uniform Mutation Range $= 0.05\times$(VarMax-VarMin); $m = 2$;		
Gravitational Search Algorithm (GSA) [45]	Year: 2009	Type: Metaheuristic	Single population
	Inspiration: Based on the law of gravity and the notion of mass, interactions and information exchange by gravitational force.		
	Parameter values: Elitist check rate $= 1$; R-power $= 1$ (R-power in Equ. (7) in [45]) ; R-norm= 2 (R-norm in Equ. (8) in [45])		
Teaching Learning Based Optimization (TLBO) [24]	Year: 2011	Type: SIA	Single population
	Inspiration: Simulates the teaching-learning process of the class room		
	Parameters: Parameters free algorithm		
Bat Algorithm (BA) [23, 43, 44]	Year: 2015	Type: SIA	Single population
	Inspiration: Echolocation behavior of simulated micro-bats, where automatic balance in exploration and exploitation of search space is by varying loudness and pulse emission rates.		
	Parameter values: Maximal and minimal pulse rates = 1, 0; Maximal and minimal frequencies = 1.5, 0; Maximal and minimal loudnesses $= 2, 1$; Gamma $= 0.2+(0.9-0.2)\times$rand; Alpha $= 0.2+(0.9-0.2)\times$rand; Frequency of updating the loudness and pulse emission rate $= 10$; Maximal and minimal probability of habitat selection $= 0.9, 0.6$; Maximal and minimal compensation rate for Doppler effect in echoes $= 0.9, 0.1$; Maximal and minimal contraction expansion coefficient $= 0.9, 0.1$; Maximal and minimal inertia weight $= 0.9, 0.5$. 'rand' is random number between 0 and 1.		
Social Learning PSO (SLPSO) [46]	Year: 2015	Type: SIA	Single population
	Inspiration: Social learning of particle swarms		
	Parameters values: c3 = nVar/PopulationSize$\times$0.01		
Binary Parallel Population Hybrid GA (BBHGA) [20]	Year: 2018	Type: EA	Multiple population
	Inspiration: Extension of GA – in each evolution epoch sampling of one crossover and selection strategy is carried out from three different methods.		
	Parameter values: $R_X =$ randomly chosen from 1, 2, and 3 in each iteration epoch. If $R_X = 1$ then Single point crossover, elseif $R_X = 2$, then two-points cross over, else for $R_X = 3$ then Uniform crossover. $R_S =$ randomly chosen from 1,2, and 3 in each iteration epoch. If $R_X = 1$ then Roulette Wheel selection, elseif $R_X = 2$ then Tournament selection, elseif $R_X = 3$ then random selection. Mutation probability $= 0.2$, $b = 20$; crossover percentage $= 0.8$; Mutation percentage $= 0.9$; Mutation rate $= 0.02$; Tournament size $= 3$.		

species. That is why at least two species are required to see the Mendelian evolution in plants.

(vi) In an evolution cycle of METO, parallel transmission of genes takes place in consecutive two generations. First genes transmission take place from F0 to F1 and then F1 to F2. These genes appear in the next generation or descendents and are called Dominant Genes (DG). The second genes transmission is in alternative generations, instead of next. These genes are recessive genes (RG) and transmit from F0 to F2 generation.

(vii) Because recessive genes are transmitting in alternative generations (F0 to F2, without appearing in F1), thus subjected to the mutation multiple times over an evolution cycle. It resembles the rehabilitation process of nature in self organizing.

In the presented manuscript, Section 2 gives the biological phenomena behind the development of the optimizer. Section 3 presents the modeling of the biological metaphors as operators and illustrates the proposed METO algorithm in steps. Section 4 shows how points move in phenotype due to operation in its genotype search space. Experimental evaluation and simulation results with limitations unveiled in Section 5. Moreover, the effect of parameters on METO performance is discussed in the same section. Section 6 exhibits future research scope for further advancement of METO. Finally, the conclusion is derived based on comparative results and statistical analysis of the METO with other optimizers.

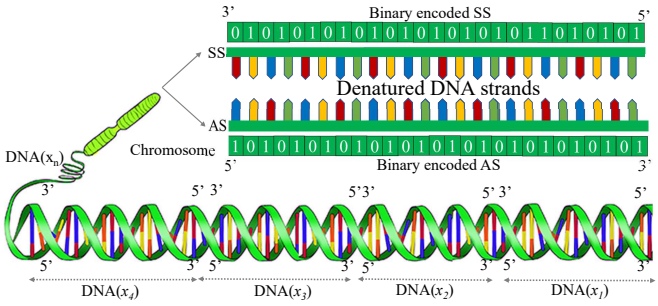


Fig. 1: Each of DNAs in a chromosome corresponds to a variable: $x_1, x_2, \ldots$

2 METO: the biological inspiration

Three concepts from genetics are adopted and redefined in order to design a meta-heuristic evolutionary search technique. It would be worth to mention here that the chromosome is typically the end to end lined up construction of many DNAs, which would run millions of miles long, as shown in Fig. 1. Following this structure of the chromosome, the *first* concept is the "denaturation and annealing of DNA" of two breeder species[2] to produce the hybrid offspring as shown in Fig. 2 [33]. As a result of natural breeding of anti-parallel strands of DNA of breeder parents are first denatured in Sense strand (SS) and Antisense strand (AS). Thereafter, SS of one breeder parent is annealed with the AS of the other to form DNA of new offspring. This concept can equally be applied on chromosomes for extracting the two strands of lined up DNAs, moreover, in Fig. 1, we can observe that $\acute{5}$ the end of one DNA is connected to $\acute{3}$ the end of other DNA. In this way, multiple lined up SS DNAs are considered as $\acute{3}-\acute{5}$ SS chromosome strand. Similarly, lined up AS DNAs are considered as $\acute{5}-\acute{3}$ AS chromosome strand. In the context of METO, the length of chromosome strand depends on the number of variables N_v in the problem space, which is equal to the number of DNAs in a chromosome.

Second concept is the evolutionary theory of Mendel [54], which revels the transmission of heredity characteristics from generation to generation as a result of breeding the parents of different species. He traced the transmission of heredity characteristics by sequential cross-breeding and self-breeding to produce F1 and F2 offspring generations, respectively. He came-up with the phenomena that (1) the dominance genes appear in the successive generations offspring such as from F0 to F1 generation offspring and from F1 to F2 generation offspring, and (2) the recessive genes appear in the second generation offspring such as from F0 to F2 generation offspring. This genetic heredity transmission from generation to generation is thought the "selfies microbes", coined by Dowkins in 1976 [51].

Third concept is biological Epimutation which guides the evolution in the organism by following the cycles of nature self-organizing behavior of sequential mutation and rehabilitation [36, 55]. This phenomena is cemented by the Joseph Heitman statement [55] that the Epimutation is a reversible phenomenon, and due to this, it results in the flexibility of the organism by giving the abilities of maintains, self-improvement, and recovery of physical strength, cognition, and mobility. In one word, it leads to the 'rehabilitation'.

A long time heredity characteristics are subjected to change due to environmental factors, which may appear as either good or bad. Good mutation[3] is acceptable by the organism as evolved traits, however, for bad mutation[4] organism responds to recover himself. Nature has blessed the organism by this self-healing capability through which an organism tries to rehabilitate itself to some extent. Moreover, pollination resembles the selection of breeder plants from different species for fertilization.

Here, we briefly explain the biological terminology and the technical inspired equivalent tools for the sake of clarity.

[2] we interchangeably use population for species

[3] Represented in METO by improved fitness of the heredity characteristics.

[4] As fitness of heredity degrades from current state.

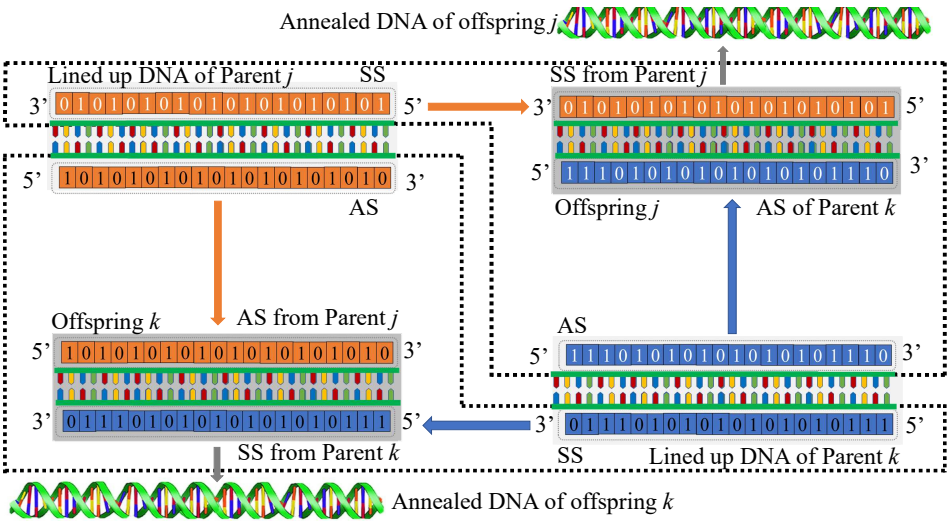


Fig. 2: Denaturing and annealing of breeder DNAs to produce the offspring

As mentioned earlier, the Mendel evolution theory came up by breeding the parents from different species. This is the base of METO to have multiple populations corresponding to distinct species. Biologically, an organism or plant can be represented by a chromosome. Each population/species contains multiple chromosomes. A chromosome structure is composed of multiple DNAs. In our implementation, each DNA is made by two binary strings which together correspond to an optimization variable. Each binary bit in this string corresponds to a genes. Each chromosome is an alternative solution which contains all the variables of the problem under-work as DNAs.

3 METO: the implementation

In this manuscript, we have established METO algorithm based on binary bit compositions being benefited from their hardware-friendly operations as searches the solution in Genome space. To implement the above described biologically inspired phenomena METO deploys four operators as *(i)* the Flipper [33], *(ii)* the Pollination [34], *(iii)* the Breeding [52], and *(iv)* the Epimutation [35, 36]. This section illustrates the implementation of above operators. Before describing the operators, encoding and decoding schemes are presented here to implement the strands of the chromosome. This is the base of METO on which all operators work.

3.1 Binary representation of the chromosome strand

In the genotype, $\mathbb{G}$, representation of the plant[5], each gene in the particular strand of the chromosome, considered as lined up DNAs, corresponds to the visible appearances, for example, height, type, size and color of the flower etc. Change in the gene value changes the above characteristics and thus results the movement in genome search space.

Implementation of the genotype $\mathbb{G}$ representation of both lined-up DNA strands is a sequence of binary bits $g[l] \in \{0,1\}$, Fig. 3, where each bit is entitled for a genes and its value for alleles. Here, l is the bit "locus" equivalent to the position of the corresponding gene $g[l]$ in the chromosome. Decoded value of $\mathbb{G}$ refers to the genetic contribution to the phenotype, $\mathbb{R} = f(\mathbb{G})$. A certain pattern of binary bits $g[l]$ in the chromosome string encompasses specific information and decoded vale of it represents a particular point in phenotype space. Thus, METO requires a coding-decoding system to represent a binary-coded string in corresponding state in real space as shown in Fig. 3. This Figure illustrates the relations of genotype representation for a particular point in phenotype space.

In Fig. 3, the one strand of chromosome is shown to represent variables, where each ten bits (genes) code one variable. This figure shows the corresponding phenotype appearance of the genotype representation of two variables x_1 and x_2. It would be worth mentioning here that the number of genes representing a variable in the chromosome strand is determined according to the desired solution accuracy. For example, in the case of two variables x_1 and x_2 with lower and upper bounds, represented by subscript L and superscript U respectively, if each contains ten genes, the lower and upper bound of the variables are illustrated by the chromosomes $(0000000000, 0000000000)$ and $(1111111111, 1111111111)$ which respectively map to (x_1^L, x_2^L) and (x_1^U, x_2^U) in real space bounds. Between the above specified limits, all intermediate chromosomes in $\mathbb{G}$ to represent points in $\mathbb{R}$ can be produced by changing one or more respective bits at any

[5] We use plant interchangeably for organism which is a more general term.

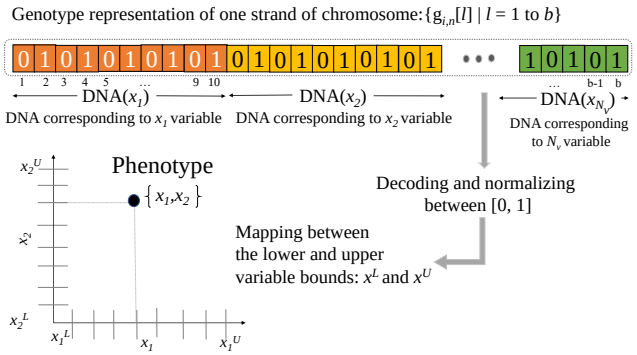


Fig. 3: Genotype and phenotype representation of a point

locus using the mapping rule $x_d = x_d^L + \frac{x_d^U - x_d^L}{2^{l_d}-1}\hat{x}_d$. Where $\hat{x}_d$ is the decoded value of binary string and is calculated as $\hat{x}_d = \sum_{d=l^d_{\text{start}}}^{l^d_{\text{end}}} 2^d g[d]$ and l_d is equal to $l^d_{\text{end}} - l^d_{\text{start}}$. In the lined up DNA strands (chromosome), the first and last bits/genes location of each DNA strand are as:

$$\begin{aligned} l^d_{\text{start}} &= (d-1)\times N_b + 1 \\ l^d_{\text{end}} &= (d \times N_b) \end{aligned} \tag{1}$$

For the l number of bits to represent a variable in the chromosome, there are 2^l possible distinct sub-strings. Thus the accuracy error in the real search space is $\frac{1}{2^l}$ [20]. However, the usual number of bits N_b to represent a variable is calculated as $N_b = \min[10,\ \text{round}(0.1 \times (x^U - x^L))]$.

3.2 Population structure

R1.1, R1.3** As Mendelian theory of evolution is based on the breeding of two species parents, the proposed algorithm is a multi-population optimizer wherein each i^{th} population $\mho_i$ is defined as

$$\mho_i = \begin{bmatrix} \mho_i^{SS} \\ \mho_i^{AS} \end{bmatrix} \tag{2}$$

Following the denaturing of DNA for breeding, population or organisms/plants $\mho_i$ can be represented by two sub-populations of all sense strand (SS) and anti-sense strand (AS) respectively as $\mho^{SS}$ and $\mho^{AS}$. Each sub-population has n number or individuals, which we interchangeability call as organisms, plants, and chromosome (lined up DNAs) as follows:

$$\begin{aligned} \mho_i^{SS} &= \{r_{i,n}\} \\ \mho_i^{AS} &= \{v_{i,n}\} \end{aligned} \tag{3}$$

Each $r_{i,n}$ and $v_{i,n}$ chromosomes strands are defined by the lined up DNA strands as shown in Figure 1.

$$\begin{aligned} r_{i,n} &= [r^1_{i,n} || r^2_{i,n} || \cdots || r^d_{i,n}] \\ v_{i,n} &= [v^1_{i,n} || v^2_{i,n} || \cdots || v^d_{i,n}] \end{aligned} \tag{4}$$

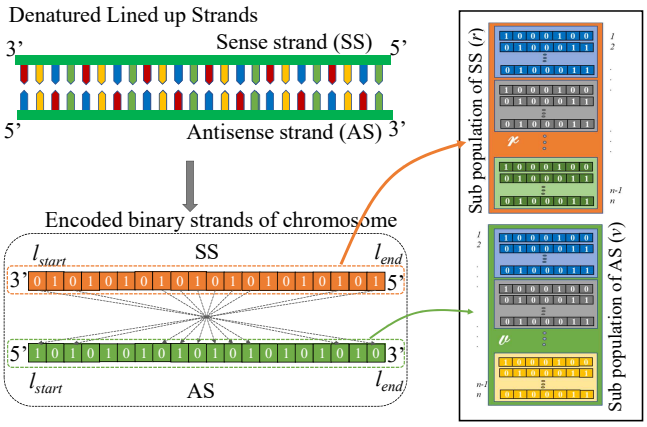


Fig. 4: Breeder Population of denatured SS and AS strands

Here $||$ is the concatenation operator, which joins the two DNA strands to form a lined up DNA or a chromosome, as shown in Fig. 1 and $d \in \{1,2,\cdots,N_v\}$. Thus, in our binary formulation, $r_{i,n}$ and $v_{i,n}$ chromosome strands are the composition of binary bits of length $b = N_b \times N_v$ and defined as:

$$\begin{aligned} r_{i,n} &= \{g^r_{i,n}[l] = \{0,1\} \quad | \quad l = 1,2,...,b\} \\ v_{i,n} &= \{g^v_{i,n}[l] = \{0,1\} \quad | \quad l = 1,2,...,b\} \end{aligned} \tag{5}$$

Here N_v and N_b represent the number of variables/DNAs and number of bits to represent a variable, respectively. The binary value of each gene $g^r_{i,n}[l]$ is randomly initialized either 0 or 1 and then evolved by the proposed algorithm. First and last bits of the strands are respectively represented by l_{start} and l_{end} and are assigned as per the the double strand structure of DNA in Fig. 1 and 4 as follows:

$$\begin{aligned} g^r_{i,n}[l_{\text{start}}] &\longleftarrow \acute{3} & g^r_{i,n}[l_{\text{end}}] &\longleftarrow \acute{5} \\ g^v_{i,n}[l_{\text{start}}] &\longleftarrow \acute{5} & g^v_{i,n}[l_{\text{end}}] &\longleftarrow \acute{3} \end{aligned} \tag{6}$$

In above formulation, d^{th} anti-sense strand (AS) is opposite composition of corresponding SS. It is defined by the Flipper operator $\mathscr{F}$ as shown in Fig. 4. Section-3.4.1 defines $\mathscr{F}$ operator in detail.

$$v^{N_v}_{i,n} = \mathscr{F}(r^{N_v}_{i,n}) \tag{7}$$

In the experiment, Mendel traced the heredity transmission in two successive generations, which are $F1$ and $F2$, where current parent population is $F0$. Produced offspring in $F1$ generation are the result of cross-breeding of $F0$ generation parents , where two parents breeder are from different species. $F2$ generation offspring are produced by the self-breeding, where one parent breeds by itself and recessive heredity $\mathscr{H}$ appears in offspring. The n^{th} strand of $\mho_k^{SS}$

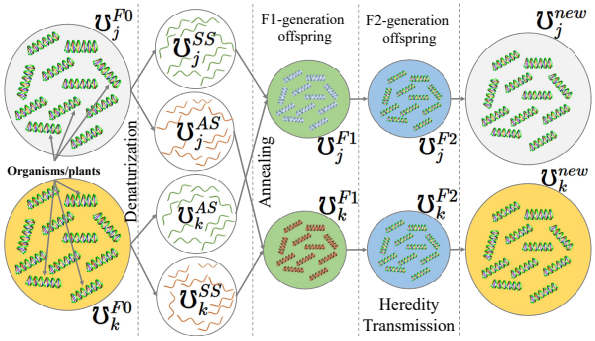


Fig. 5: Illustration of production of $F1$ and $F2$ generation offspring and new population

breeds with n^{th} strand of $\mho_j^{AS}$ to produce $F1$ generation offspring strand $r_{k,n}^{F1}$ for k^{th} species (see Fig. 5). The populations in an evolution epoch are defined as

$$\begin{aligned} &\mho_j^{F0} \rightarrow \mho_j^{F1} \rightarrow \mho_j^{F2} \rightarrow \mho_j^{new} \\ &\mho_k^{F0} \rightarrow \mho_k^{F1} \rightarrow \mho_k^{F2} \rightarrow \mho_k^{new} \end{aligned} \tag{8}$$

Here, $|\mho_j^{F1}| \leq |\mho_j^{F0}|$ and $|\mho_j^{F2}| \leq |\mho_j^{F1}|$ where $|.|$ indicates the size of population. Due to following the elitism, best individuals are selected to breed the offspring, thus, new offspring contains the less number of genes but from the best characteristic of parents. Note besides the above inequality, the resultant population size of the epoch $\mho_j^{\text{new}}$ is acquired in a way to be same size as $\mho_j^{F0}$ size by acquiring the SS strands of $\mho_j^{new}$ and $\mho_k^{new}$ population as follows

$$\begin{aligned} r_{j,n}^{\text{new}} &= \text{best}\{r_{j,n}^{F0} \in \mho_j^{F0},\ r_{j,n}^{F1} \in \mho_j^{F1},\ r_{j,n}^{F2} \in \mho_j^{F2}\} \\ r_{k,n}^{new} &= \text{best}\{r_{k,n}^{F0} \in \mho_k^{F0},\ r_{k,n}^{F1} \in \mho_k^{F1},\ r_{k,n}^{F2} \in \mho_k^{F2}\} \end{aligned} \tag{9}$$

where $r_{j,n}^{\text{new}}$ and $r_{k,n}^{\text{new}} \in \mho_j^{new}$. Note that the developed algorithm is based on heredity transfer from one generation to the next. This heredity can only be transferred through the parents to the successive offspring. Thus the parents are subjected to change by evolved offspring of the same location. In other words, n^{th} location parent is changed only by the n^{th} location offspring. In this manner, the heredity of n^{th} parent transmits to the corresponding offspring properly. The resulting new parent population for i^{th} species $\mho_i^{\text{new}}$ is as follows:

$$\mho_i^{\text{new}} = \begin{bmatrix} r_i^{\text{new}} \\ \mathscr{F}(r_i^{\text{new}}) \end{bmatrix} \tag{10}$$

3.3 Construction of heredity

This section illustrates the construction of recessive genes (RG), which is an important part of the algorithm. According to the Mendelian evolution theory, heredity transmits

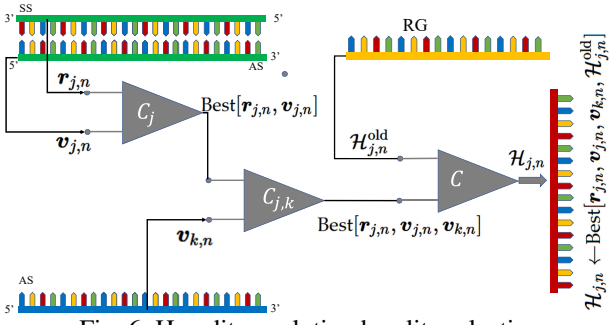


Fig. 6: Heredity evolution by elite selection

from generation to the next. According to his theory, two types of heredity exist; Dominant Genes (DG) and Recessive Genes (RG). DG appear in the $F1$ generation offspring, and RG are intended to appear in the $F2$ generation offspring. Thus, we retain RG as the heredity $\mathscr{R}$, to be available for $F2$ generation offspring. Fig. 6 shows the construction of heredity genes, $\mathscr{H}$ by the elit selection of genes from one generation to next.

After cross-breeding, two breeder parents $r_{j,n}$ and $v_{k,n}$ have their own recessive heredity information. Since we are developing an evolutionary algorithm, thus always best fitness genes are intended to pass to the next generation – implicit elitism property [53]. This process is a elite selection of heredity and accomplished by three comparators $\mathscr{C}_j$, $\mathscr{C}_{j,k}$, and $\mathscr{C}$ as shown in Fig. 6. $\mathscr{C}_j$ selects the best from $[r_{j,n}, v_{j,n}]$ based on their fitness. The output is then compared with $v_{k,n}$ using the comparator $\mathscr{C}_j$, which provides the best heredity from $[r_{j,n}, v_{j,n}, v_{k,n}]$. The output goes to the comparator $\mathscr{C}$ for comparison with the reference old heredity $\mathscr{H}_{j,n}^{\text{old}}$. Then the final selected heredity $\mathscr{H}_{j,n}$ is available for the n^{th} $F2$ generation offspring of j^{th} species. Similarly, $\mathscr{H}_{k,n}$ is produced for the n^{th} $F2$ generation offspring of k^{th} species.

$$\begin{aligned} \mathscr{H}_{j,n} &\leftarrow \text{best}[r_{j,n}, v_{j,n}, v_{k,n}, \mathscr{H}_{j,n}^{\text{old}}] \\ \mathscr{H}_{k,n} &\leftarrow \text{best}[r_{k,n}, v_{k,n}, v_{j,n}, \mathscr{H}_{k,n}^{\text{old}}] \end{aligned} \tag{11}$$

3.4 Basic operators

METO ensembles biologically inspired phenomena in the form of four operators to accomplish the four different tasks as described in this section.

3.4.1 Flipper operator ($\mathscr{F}$)

In the way of creating AS from SS, Flipper operator, $v_{i,n} = \mathscr{F}(r_{i,n})$, is introduced which reverse the order of bits in SS, see Fig. 4. It is inspired from the natural representation of double strand DNA structure, where, AS strand and SS strand are opposite in nature and respectively represented by 5́–3́

Algorithm 1 Producing AS using Flipper Operator

Require: r_i, SS offspring sub-population of i^{th} species, N_v and N_b are respectively number of variables and associated number of bits to form a DNA, N is number of individuals in i-th species, δ_{N_b}, δ_{N_v} are the bits flipping rate and DNAs selection rate for flipping, respectively.

function FLIPPER($r_i, N_v, N_b, \delta_{N_b}, \delta_{N_v}$)
  $v_i \leftarrow r_i$ ▷ Copy template from heredity
  $F_i \leftarrow zeros(size(v_i))$ ▷ Initialize the flipped strands
  $\delta_{N_v} \leftarrow \lceil \text{rand} * N_v \rceil$ ▷ Number of DNAs for flipping
  $d \leftarrow$ Randomly choose δ_{N_v} DNAs from N_v DNAs
    for $n \leftarrow 1$ to size of sub-population r_i **do**
      for $\forall\, i \in d$ **do**
      $l^d \leftarrow (i-1)N_b + 1$ to $i * N_b$) ▷ DNA bits
        for $\forall\, l \in l^d$ **do**
          $\delta_{N_b} \leftarrow$random number between 0 and 1
            if $\delta_{N_b} \leq \delta$ **then** ▷ δ is from equation (14)
            $v_i(n,l) \leftarrow \mathscr{F}(r_i(n,l))$) ▷ equation(12)
            end if
          end for
      end for
    end for
    return v_i
end function

and 3́–5́, as in Fig. 2 and 4: Gene $g^v_{i,n}[l] \in v_{i,n}$ at location l is obtained by Flipper operator $\mathscr{F}(.)$ and is defined to reverses the order of $g^r_{i,n}[l]$ bits as:

$$g^v_{i,n}[l] = \mathscr{F}(g^r_{i,n}[l]) = g^r_{i,n}[\hat{l}] \tag{12}$$

To introduce the stochastic nature in the Flipper operator, a Binary variable $X \in \{0,1\}$ is defined, based on which the equation (12) puts the $\hat{l} = |l_{\text{end}} \times (1-X) - l + 1 - X|$ location bit to l position, where $|.|$ denotes absolute value. If $X = 1$, the bits do not flip due to $|\hat{l}| = l$, while $X = 0$, $\hat{l} = l_{\text{end}} - l + 1$ which results in flipping the bits. Thus, in equation (12), only those bits of $\mathscr{F}(r_{i,n})$ will be flipped, for which $|\hat{l}| \neq l$.

Flipper operator is essential in initial evolution epochs of the algorithm since it tries to explore the maximum problem space and reduces the risk of being trapped at premature convergence or local extremes. In the long run, this affects the solution accuracy and oppose the exploitation concept, thus need to control as evolution epochs grow. Thus, flipping probability δ is introduced as a control mechanism which controls the binary variable X as in equation (12):

$$X = \begin{cases} 0 & : \text{if } \delta_{N_b} \leq \delta \\ 1 & : \text{otherwise,} \end{cases} \tag{13}$$

δ_{N_b} is a randomly generated number, and δ is the function of maximum number of maximum function evaluations (α) and the current function evaluation (β). It is defined as an exponentially decreasing function:

$$\delta = \max[0.1,\ \min[0.7,\ (1.1 - \frac{0.1}{N_b}) \times exp(-(3 + \frac{1}{N_b}) \times \frac{\beta}{\alpha})]] \tag{14}$$

Moreover, inspired from the error in DNA replication, we introduced the mechanism where Flipper operator applied on the selected DNA from N_v DNAs as per equation (15). Candidate DNSs are selected randomly by the factor δ_{N_b}.

$$\delta_{N_b} = N_v \times \lceil N_v * rand \rceil \tag{15}$$

where $\lceil . \rceil$ is the ceiling operation. To control the search strategy, Flipper operator is applied on selected DNAs δ_{N_v} out of N_v in the chromosome strand, based on the random number rand $\in [0,1]$. Pseudocode for producing AS by Flipper operation is given in Algorithm 1.

3.4.2 Pollination operator

In the METO algorithm, two pollination schemes are used in sequence as called here random and sequential. First, the random pollination randomly picks the two breeder species/population from multiple even number of species as $\mho_j$ and $\mho_k$. Then, in sequential pollination, each individual of $\mho_j$ is paired in bijection with an individual of $\mho_k$ at the same order.

$$[\mho_j, \mho_k]_{j \neq k} = \mathscr{P}(\mho) \qquad \mho = \{\mho_i \mid i = 1, 2, \cdots, M\} \tag{16}$$

where, $\mho$ is set of all populations, $\mathscr{P}(.)$ is pollination operator, and M is the number of species. One can notice that two selected populations j and k are not equal. Sequential pollination is the simplistic one, which can be extended to the other pollination schemes such as Roulette Wheel, Tournament and etc. [16].

3.4.3 Breeding operators

To implement the Mendelian theory, two breeding operators *Cross-breeding* and *Self-breeding* are defined to produce the two successive generations, $F1$ and $F2$, offspring, respectively. The definition of both breeding operators are as follows:

Cross-breeding

To produce the n-th $F1$ generation offspring, breeder parents chromosome strands $r_{j,n}$ and $v_{k,n}$ are selected from different species j and k, using pollination operator. Cross-breeding operator $\mathscr{B}(.)$ produces the SS for the species corresponding to $r_{j,n}$, where the corresponding AS for the same species is constructed by $\mathscr{F}(.)$ operator. AS is treated as a supporting strand to produce SS, because in genetics, SS is coding strand. Operator $\mathscr{B}(.)$ produces twin SS offspring

$$\left[r^{F11}_{j,n}, r^{F12}_{j,n}\right] = \mathscr{B}(r_{j,n}, v_{k,n}) \tag{17}$$

$$\left[v^{F11}_{j,n},\ v^{F12}_{j,n}\right] = \left[\mathscr{F}(r^{F11}_{j,n}), \mathscr{F}(r^{F12}_{j,n})\right] \tag{18}$$

$$\mho^{F11}_{j,n} = \left[r^{F11}_{j,n}, v^{F11}_{j,n}\right]^T \ :\ \mho^{F12}_{j,n} = \left[r^{F12}_{j,n}, v^{F12}_{j,n}\right]^T \tag{19}$$

Algorithm 2 F1 generation offspring

Require: r_j and v_k are the set of Sense and Anti-sense strands of different species j^{th} and k^{th}, respectively; N_v and N_b are respectively number of variable and number of bits to represent each variable

function F1-OFFSPRING(r_j, v_k, N_v, N_b)
$p \leftarrow$ size of sub-population r_j ▷ number of plants available for cross-breeding, where size of r_j = size of v_k
  $r_{11} \leftarrow r_j$ ▷ initializing first offspring with the genes of r_j
  $r_{12} \leftarrow r_j$ ▷ initializing second offspring with the genes of r_j
  $f1 \leftarrow$ false Boolean matrix of dimension $p \times (N_v \times N_b)$
  for $j \leftarrow p$ **do**
  $d \leftarrow$ randomly select $\lceil N_v \times rand \rceil$ variables from N_v
    for $\forall i \in d$ **do** ▷ for all elements in vector d
    $l^d \leftarrow (i-1)N_b + 1$ to $i * N_b$ ▷ N_b bits corresponding to i^{th} variable or DNA strand in biological term
    $f1(j, l^d) \leftarrow$ true ▷ Change all N_b bits associated with i^{th} variable to 0

    end for
  end for
$f2 \leftarrow \neg f1$ ▷ Boolean matrix $f2$ and $f1$ have opposite bits at the same place
$r_{11}(f1) \leftarrow v_k(f1)$ ▷ transferring genes of v_k to 1^{st} offspring
$r_{12}(f2) \leftarrow v_k(f2)$ ▷ transferring genes of v_k to 2^{nd} offspring
  return r_{11}, r_{12} ▷ Twin offspring
end function

Algorithm 3 F2 generation offspring

Require: F_1, F1 generation offspring; $\mathscr{H}$, heredity; $f_{\mathscr{H}}$, fitness of $\mathscr{H}$; f_{F_1}, fitness of F_1 ; ρ^L, and ρ^U, lower and upper limit of Mendelian probability ρ, respectively
function F2-OFFSPRING(F_1, $\mathscr{H}$, $f_{\mathscr{H}}$, f_{F_1} ρ^L, ρ^U)
$[s_1, s_2] \leftarrow$ size of F_1 ▷ Subscript '1' and '2' represents respectively for dimension 1 and dimension 2
$F_2 \leftarrow$ false(s_1, s_2) ▷ Initialization of the F2 generation offspring of size F_1
  for $i \leftarrow s_1$ **do**
$F \leftarrow$ chromosomes from F_1, for which $I_{\mathscr{H}} = 1$
$H \leftarrow$ all chromosomes from $\mathscr{H}$, for which $I_{\mathscr{H}} = 1$
$I \leftarrow$ false$(1, s_1)$ ▷ Initialization of another logical index to represent genes/bits
$B \leftarrow (F \neq H)$ ▷ True value '1' in B represents that corresponding bits in chromosome F and H, which are not equal
$s \leftarrow$ sum of all true values in B
$r \leftarrow$ generate random numbers between $[0,1]_{1\times s}$
$\rho \leftarrow$ generate random numbers between$[\rho^L, \rho^U]_{1\times s}$
$I(B_n) \leftarrow [r < \rho]$
$r \leftarrow$ (sum of "true" in all rows of I) > 0 ▷ 1 in r Boolean vector is representing updated individual
    if $f_{\mathscr{H}}(i) < f_{F_1}(i)$ **then** ▷ Heredity is dominating on the current F1 generation genes
    $F(I) \leftarrow H(I)$ ▷ Transfer heredity Because they are better than the F1 generation offspring genes
    $F_2(i,:) \leftarrow F$ ▷ Place new offspring in F_2 population
    end if
    if $f_{F_1}(i) < f_{\mathscr{H}}(i)$ **then**
    $H(I) \leftarrow F(I)$ ▷ Heredity genes are not dominating well due to weaker fitness than F1 offspring genes
    $F_2(i,:) \leftarrow H$ ▷ Place new offspring in F_2 population
    end if
    if $f_{\mathscr{H}}(i) == f_{F_1}(i)$ **then**
    production of F2 generation offspring is void for i^{th} F1 generation offspring, because of same fitness of genes.
    end if
  end for$F_2 \leftarrow$ remove all chromosomes from F_2 population which have all bits = '0'
  return F_2
end function

Following the stochastic approach for selecting the dominant genes from two breeder parents $r_{j,n}$ and $v_{k,n}$, natural selection is adopted. A variable y is defined for cross-breeding, where it has d elements. Each element in y may have either 1 or 0 as per the generated random number γ in the range of $[0, 1]$:

$$y[d] = \begin{cases} 1 & \gamma \geq 0.5 \\ 0 & \text{otherwise,} \end{cases} \tag{20}$$

Twin offspring SS chromosomes are then generated by the following equations:

$$r_{j,n}^{F11}[l_{\text{start}}^d : l_{\text{end}}^d] = \begin{cases} r_{j,n}[l_{\text{start}}^d : l_{\text{end}}^d] & y[d] = 1 \\ v_{k,n}[l_{\text{start}}^d : l_{\text{end}}^d] & \text{otherwise,} \end{cases} \quad (21)$$

$$r_{j,n}^{F12}[l_{\text{start}}^d : l_{\text{end}}^d] = \begin{cases} v_{k,n}[l_{\text{start}}^d : l_{\text{end}}^d] & y[d] = 1 \\ r_{j,n}[l_{\text{start}}^d : l_{\text{end}}^d] & \text{otherwise,} \end{cases} \quad (22)$$

From the twin offspring, we select best of them, based on their fitness, as the resulting $F1$ generation offspring SS.

$$r_{j,n}^{F1} = \text{best}[r_{j,n}^{F11}, r_{j,n}^{F12}] \quad (23)$$

However, multiple offspring sense strands can be produced as shown in Fig. 7. Both twin SS offspring $[r_{j,n}^{F11}, r_{j,n}^{F12}]$ is composed of dominant genes either from $r_{j,n}$ or $v_{k,n}$ based on natural selection.

In a similar manner, the offspring are produced for S_k species where sense and anti-sense parents chromosomes are respectively $r_{k,n}$ and $v_{j,n}$. For generating multiple twin offspring, the common genes can be obtained in advance using XNOR logical process, and for the remaining locus of the genes, the above equation is used. In the case of very long chromosomes, this can help to decrease the time complexity for producing multiple offspring.

$$\text{XNOR}(r_{j,n}, v_{k,n}) = r_{j,n}[l] \wedge v_{k,n}[l] + \bar{r}_{j,n}[l] \wedge \bar{v}_{k,n}[l], \quad (24)$$

where $\bar{\cdot}$ indicates the complement of the bit. Transfer of common characteristics has been shown in Fig. 7, where gray genes show the common ones at 3^{rd} and 4^{th} locus as a result of $XNOR$ operator, where others are undefined genes and represented by allele value X.

Self-breeding

The self-breeding operator $\mathscr{O}$ breeds the SS strands of offspring generated in $F1$ generation with itself to produce the $F2$ offspring SS for j^{th} species. This is the process which

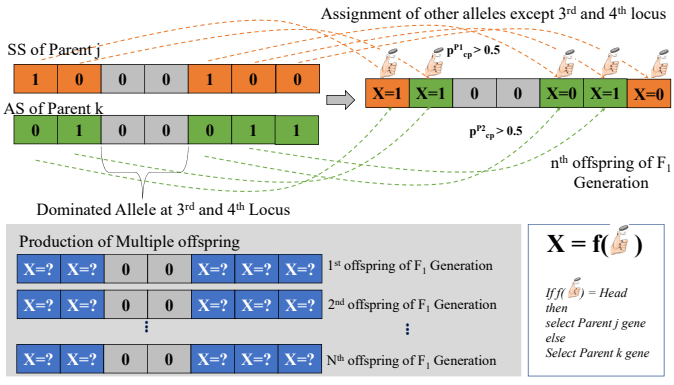


Fig. 7: Illustration of production of $F1$ generation offspring

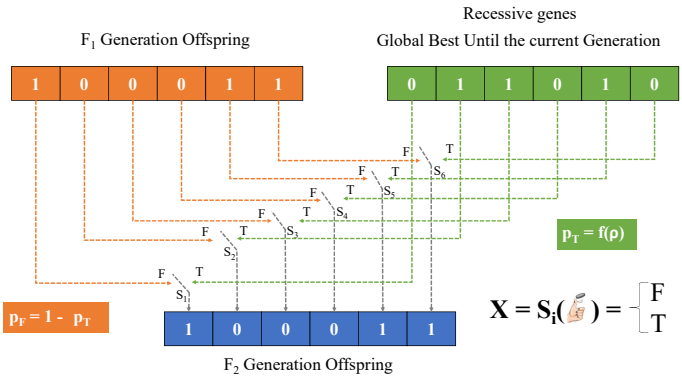


Fig. 8: Illustration of production of $F2$ generation offspring

makes dominant the recessive heredity $\mathscr{H}_{j,n}$ with Mendelian probability ρ in the offspring.

$$r_{j,n}^{F2} = \mathscr{O}(r_{j,n}^{F1}, r_{j,n}^{F1}) \quad (25)$$

RG dominates in $F2$ offspring by replacing the same locus genes of $F1$ generation offspring If the fitness of RG is better than $F1$ generation offspring, which is parent for $F2$ generation offspring. On the other hand if fitness of RG is not better than $F1$ generation offspring, then genes of $F1$ generation offspring will be dominating in $F2$ generation offspring. It is controlled by the random number ρ in the range of $[\rho^L, \rho^U]$. Generally, $\rho^L = 0.9$ and $\rho^U = 0.97$ are selected. The effect of different values of ρ on the search strategy is discussed in the Section 11 and in Fig. 4.

$$\mathscr{O}(r_{j,n}^{F1}, r_{j,n}^{F1}) = \begin{cases} r_{j,n}^{F2}[l] \leftarrow \mathscr{R}_{j,n}[l] & : \text{if } \rho^L \le \rho \le \rho^U \\ r_{j,n}^{F2}[l] \leftarrow r_{j,n}^{F1}[l] & : \text{otherwise} \end{cases} \quad (26)$$

Similarly, $r_{k,n}^{F2}[l]$ for k species are produced by $\mathscr{O}(r_{k,n}^{F1}, r_{k,n}^{F1})$ as in equation 26. Offspring in $F2$ generation is denoted as $\mho^{F2}$, where AS strand is generated by the Flipper operator $\mathscr{F}$. Thus, the population of $F2$ generation offspring for j and k species are as

$$\mho_j^{F12} = \left[r_j^{F2}, v_j^{F2}\right]^T \qquad \mho_k^{F12} = \left[r_k^{F2}, v_k^{F2}\right]^T \quad (27)$$

Pseudocode for producing $F2$ generation offspring is given in Algorithm-3. It would be interesting to observe that in line number 14 and 18 of Algorithm-3 that if heredity fitness of the $F1$ generation offspring is better then heredity is transferred to produce $F2$ generation offspring. On the other hand, If fitness of $F1$ generation offspring is better than its heredity, then $F1$ offspring genes dominated to produce $F2$ generation offspring. This transfer of genes to the $F2$ generation offspring is based on the ρ, the Mendelian probability.

Algorithm 4 Epimutation in Heredity

Require: $\mathscr{H}$, the set of heredity of all N organism in a species; τ^L, τ^U, Lower and Upper Limits of the mutation probability; ζ, the mutation rate for selecting individuals for mutation from the population, τ_{N_v}, DNAs mutation rate, N_v, N_b

function EPIMUTATION($\mathscr{H}, \tau^L, \tau^U, \tau_{N_v}, \zeta, N_v, N_b$)
  $s \leftarrow \mathscr{H}$ ▷ Copy binary heredity $\mathscr{H}$ vector of dimension $N \times (N_v N_b)$ to the variable s
  $\xi \leftarrow 0$ ▷ initializing counter for rehabilitation attempts
  index $\leftarrow 0$ ▷ This contains indexes of all organisms who have evolved after going through the mutation attempt
  while ($\neg$empty(ζ) && $\xi < 1$) **do** index =[]; ▷ index for evolved hereditary from all due to mutation
    for $i \leftarrow \xi$ **do** ▷ Do the process inside the loop for $i =$ each element of ζ
      $m \leftarrow \lceil \tau_{N_v} \times N_v \rceil$ ▷ Number of heredity DNA strands for evolution
      $u \leftarrow$ Selected m DNAs form N_n ▷ Each element of u is an index and represents a selected DNA strand
      for $j \leftarrow u$ **do** ▷ Do the process inside the loop for $j =$each element of u
        $b \leftarrow (j-1)N_b + 1$ to $j * N_b$) ▷ DNA strand bits from a chromosome, it selects N_b bits according to j value
        $d \leftarrow$ generate a random vector of length $1 \times N_b$ between $[\tau^L, \tau^U]$
        $r1 \leftarrow$ generate a random vector of length $1 \times N_b$ between 0 and 1
        $d \leftarrow (r1 < d)$ ▷ 1 in left side Logic vector d, shows that associated bit should mutate: $0 \Longleftrightarrow 1$
        $s(i,b) \leftarrow XOR(s(i,b), d)$ ▷ XOR mutates the bits of heredity DNA strand as per the 1s in d
      end for
      $F_s \leftarrow$ Calculate Fitness of $\bar{s}(i)$ ▷ Element of $\bar{s}(i)$ are the binary-to-decimal conversion of binary DNA strand of $s(i)$
      if $F_s \leq F_{\mathscr{H}}(i)$ **then** ▷ $F_{\mathscr{H}}(i)$ is the fitness of $\mathscr{H}(i)$ heredity strands
        $\mathscr{H}(i) \leftarrow s(i)$ ▷ Replace existing heredity by evolved heredity
        index $=$ [index, i] ▷ Store the index of all evolved heredity strands
      end if
    end for
    $D \leftarrow$ remove(D, index) ▷ remove evolved organism from the D
    if isempty(index) **then**
      $\xi \leftarrow \xi + 1$ ▷ increase the termination counter by 1, if no organism is evolved
    end if
  end while
  return $\mathscr{H}$ ▷ Output is the evolved heredity
end function

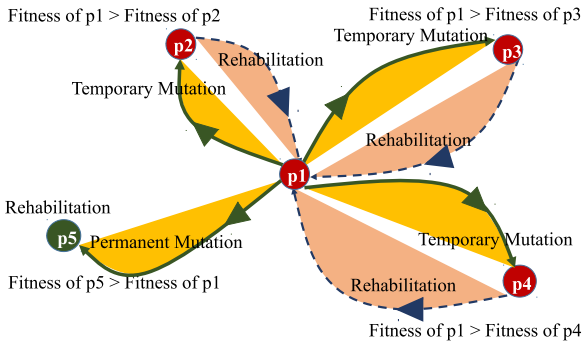


Fig. 9: The Epimutation process

3.4.4 Epimutation operator

Fig. 9 shows the Epimutation process of the heredity of an organism where the five states of the organism are shown as p1 p2, p3, p4 and p5. Here p1 is the current state of the organism, and others are the results of Epimutation. p2, p3 and p4 states are not better than the current state; thus organism returns back to its original state p1 by a rehabilitation process. But, mutated state p5 is better than the current state; thus, in this case, the organism accepts it as evolution. This mechanism is Epimutation and tries to find the better state of the pseudo-global optima, and we call it – fine-tuning – of the global best point by iterative mutations followed by rehabilitation. We define Epimutation factor ξ, which gives the number of chances to the organism to go through the mutation in their life cycle. Here, the organism took four attempts to get a better solution, thus $\xi = 4$.

$$\mathscr{H} \leftarrow \mathscr{E}(\mathscr{H}) = \mathscr{E}(g^{\mathscr{H}}[l]) \tag{28}$$

$\mathscr{E}$ operator changes the genes value of heredity $g^{\mathscr{H}}[l]$ based on the randomly generated Epimutation factor τ. For each gene, if τ lies between the lower and upper limits then the corresponding gene will be changed according to the equation below

$$g^{\mathscr{H}}[l] = \begin{cases} 1 - g^{\mathscr{H}}[l] & : \text{if } \tau^L \leq \tau \leq \tau^U \\ g^{\mathscr{H}}[l] & : \text{otherwise,} \end{cases} \tag{29}$$

The lower and upper limits of τ is calculated as follows:

$$\begin{aligned} \tau^L &= \max\left[\frac{1}{N_b^2}, \min\left(0.2,\ (0.3 - \frac{1}{N_b})\exp(-(4 + \frac{1}{N_b})\frac{\beta}{\alpha})\right)\right] \\ \tau^U &= \max\left[\frac{1}{N_b}, \min\left(0.3,\ (0.5 - \frac{2}{N_b})\exp(-(2 + \frac{3}{N_b})\frac{\beta}{\alpha})\right)\right] \end{aligned} \tag{30}$$

It is worth to note that heredity is used to produce the $F2$ generation offspring from self-breeding of $F1$ generation parent. Thus only those heredity are subjected to Epimutaion which are associated with $F1$ generation parents, intended to self-breed.

Algorithm 5 METO Algorithm

Require: M is the number of species, N_v, N_b, N is the number of individuals, TerminationCriteria, CostFunction

▷ Iter, feval, solution accuracy are the general TerminationCriteria and Fitness if calculated based on CostFunction

for $j \leftarrow 1$ to M **do** ▷ $p \leftarrow M \times \lceil \frac{N}{M} \rceil$, each M species contains p organism/plants/individuals/DNAs

$\mho(j).r \leftarrow$ Initialize SS population randomly

$\mho(j).v \leftarrow \text{FLIPPER}\big(\mho(j).r(n), N_v, N_b, N, \delta_{N_b}, \delta_{N_v}\big)$ ▷ $\mathscr{F}$ operator to produce AS strand with $\delta_{N_b} = \delta_{N_v} = 1$

$\mho(j).\mathscr{H} \leftarrow$ best of $[\mho(j).r, \mho(j).v]$ ▷ Heredity formation, taking best of SS and AS strands based on fitness

$S_{best}(j) \leftarrow$ best of $\mho(j).\mathscr{H}$ ▷ Taking best heredity of all species, Species best (S_{best})

end for

$G_{best}(1) \leftarrow$ best of S_{best} ▷ Taking best heredity from all species, Global best (G_{best})

$i \leftarrow 2, \rho^L \leftarrow 0.9,\ \rho^U \leftarrow 0.97$ ▷ i is evolution epoch counter for while loop

while until termination criteria met **do**

for $j \leftarrow 1$ to M **do**

$k \leftarrow$ select another species based on random pollination

δ_{N_b}, τ^U and $\tau^L \leftarrow$ based on equation (14) and (30), respectively

δ_{N_v} and $\tau_{N_v} \leftarrow$ generate a random number between $[0, 1]$

$\nabla_{F1} \leftarrow \max[0.3, rand]$ ▷ Initialize cross-breeding rate that how many organisms/plants are ready for cross-breeding

$\nabla_{F2} \leftarrow \max[0.6, rand]$ ▷ Self-breeding rate: It is to select the parents organisms/plants from $F1$ generation for self-breeding

$E_{F1} \leftarrow$ index of best p_{F1} strands from population $\mho(j).r$ ▷ Implicit Elitism: organisms for cross-breeding: $p_{F1} \leftarrow \lceil p \times \nabla_{F1} \rceil$

$\mho(j).\mathscr{H}(E_{F1}) \leftarrow \text{best}[\mho(k).v(E_{F1}),\quad \mho(j).\mathscr{H}(E_{F1})]$ ▷ Due to cross-breeding heredity of two species are evolving

$\xi \leftarrow E_{F2} \leftarrow$ index of best (p_{F2}) strands from the $F1$ population ▷ For self-breeding and Epimutation, $E_{F2} \subseteq E_{F1} : p_{F2} \leftarrow \lceil |p_{F1}| \times \nabla_{F2} \rceil$ $r_{F1} \leftarrow$ Select ∇_{F1} SS strands from $\mho(j).r(E_{F1})$

$v_{F1} \leftarrow$ Select ∇_{F1} AS strands from $\mho(k).v(E_{F1})$

$[\mho 1^{F1}, \mho 2^{F1}] \leftarrow \text{F1-OFFSPRING}(r_{F1}, v_{F1}, N_v, N_b)$

$F1 \leftarrow \text{best}[\mho 1^{F1},\ \mho 2^{F1}]$ ▷ Best of two offspring from the same parents are selected based on their fitness

$[\mho(j).r(E_{F1}) \leftarrow F1]$ and $[\mho(j).\mathscr{H}(E_{F1}) \leftarrow F1]$ ▷ Replace SS strands and heredity at positions E_{F1} in main population

$I_v \leftarrow \lceil |p_{F1}| \times rand \rceil$ ▷ rand is the randomly generated number: $I_v \subseteq E_{F1}$

$v_{F1} \leftarrow \text{FLIPPER}(F1(I_v), N_v, N_b, \delta_{N_b}, \delta_{N_v})$ ▷ producing AS strands of $F1$ generation SS on I_v position

$\mho(j).\mathscr{H}(I_v) \leftarrow \text{best}[\mho(j).\mathscr{H}(I_v),\quad v_{F1}]$ ▷ Replace I_v positioned heredity

$s_{\mathscr{H}} \leftarrow$ Select ξ organism's heredity from $\mho(j).\mathscr{H}$ for Epimutation before self-breeding

$\mho(j).\mathscr{H}(\xi) \leftarrow \text{EPIMUTATION}(s_{\mathscr{H}}, \tau^L, \tau^U, \tau_{N_v}, \xi, N_v, N_b)$ ▷ This process evolve the heredity of organisms on place ξ

$r_{F2} \leftarrow$ Select ∇_{F2} strands from r_{F1}

$\mho^{F2} \leftarrow \text{F2-OFFSPRING}(r_{F2}, \mho(j).\mathscr{H}(\xi), \rho^L, \rho^U)$

$\mho(j).r(E_{F2}) \leftarrow \text{best}[\mho(j).r(E_{F2}),\quad \mho^{F2}]$ ▷ replace parents in main population, if $F2$ generation offspring fitness is better

$I_v \leftarrow \lceil |p_{F2}| \times rand \rceil$ ▷ rand is the randomly generated number: $I_v \subseteq E_{F2}$

$v_{F2} \leftarrow \text{FLIPPER}(F2(I_v), N_v, N_b, \delta_{N_b}, \delta_{N_v})$ ▷ Produce AS strands of $F2$ generation offspring on I_v position

$\mho(j).\mathscr{H}(I_v) \leftarrow \text{best}[\mho(j).\mathscr{H}(I_v),\quad v_{F1}]$ ▷ Replace I_v positioned heredity

$v \leftarrow \text{sort}([\mho(j).v;\quad v_{F1};\quad v_{F2}])$ ▷ Concatenate and sort the AS strands based on their Fitness

$\mho(j).v \leftarrow$ First best p AS strands of v

$S_{best}(j) \leftarrow \text{best}[\mho(j).\mathscr{H}]$ ▷ Taking best heredity of all species, Species best (S_{best})

end for

$G_{best}(i) \leftarrow \text{best}[S_{best}]$ ▷ Taking best heredity from all species, Global best (G_{best})

if $(G_{best}(i)) \geq (G_{best}(i-1))$ **then** ▷ Comparison is based on the Fitness

$G_{best}(i) \leftarrow G_{best}(i-1)$

end if

Terminate if one of the termination criteria met

$i \leftarrow i+1$

end while

return G_{best} ▷ In this algorithm, Fitness is calculated for each new strand for the given CostFunction(x)

▷ Each sub-segment of binary strand in Genotype representation $\mathbb{G}$, representing a variable, is converted in its equivalent real domain ($\mathbb{R}$) before placing in CostFunction(x); $\mathbb{G}(r) \longrightarrow \mathbb{R}(x)$

Epimutation is the self-organization mechanism, in which an organism self-adjust them-self against the environmental mutation via a rehabilitation process. Selfies microbes preserve the recessive genes and carry them from generation to generation. Here, we assumed that heredity RG undergoes the mutation process multiple times over the plant life cycle. If it gains a better mutation, adapts it as evolution process; otherwise, the organism rehabilitate them-self to the former situation. Epimutation – rehabilitation against mutation process – is shown in Algorithm 4 as a function.

3.5 METO algorithm

A block diagram of the processes in one evolution of the METO is presented in Fig. 10. Here, we can observe two parallel processes of evolution, the inner and the outer. The inner process ensembles the interactions of pollination, cross-breeding, self-breeding, and flipper operators to produce two successive generation offspring. Although, outer process deals with the heredity formation and its Epimutation operation, which is used to produce $F2$ generation offspring by self-

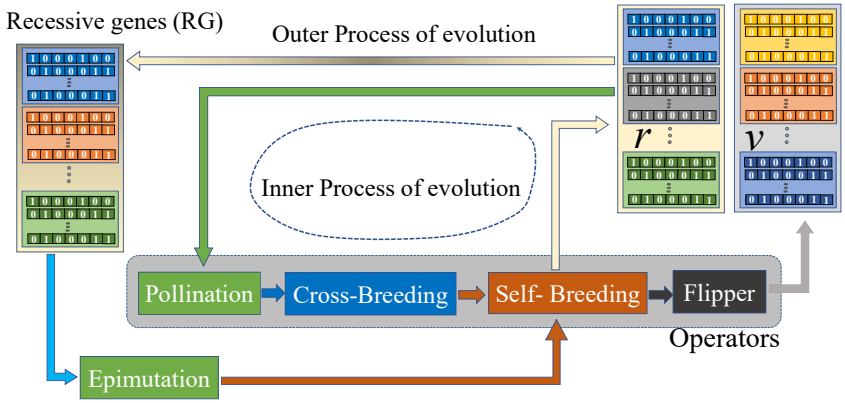


Fig. 10: Block diagram of the processes in an evolution

breeding. Flipped strands are resembling v population. The best solution acquired by each individual forms the corresponding heredity. In each evolution epoch, each species has its own best solution, which is S_{best}. At the last of evolution epoch, best of all species is extracted to form the "Global Pseudo-best (G_{best})". The term "pseudo" comes from the fact that the solutions improve their merit in being closer to the global optimal solution at each evolution. Algorithm 5 presents the pseudocode of METO as it deploys the above described four operations in the sequence. It would be worth to mention again that the current heredity is replaced with strands of $F1$ and $F2$, if they are better than current RG. METO concludes the solution, once, one of the following termination criteria met: 1) Maximum number of iterations, 2) Individuals of all species have the same heredity and not changing in N iterations, 3) The same answer is coming in each successive iteration for m times, and 4) If the error is below 10^{-5} or a desired value.

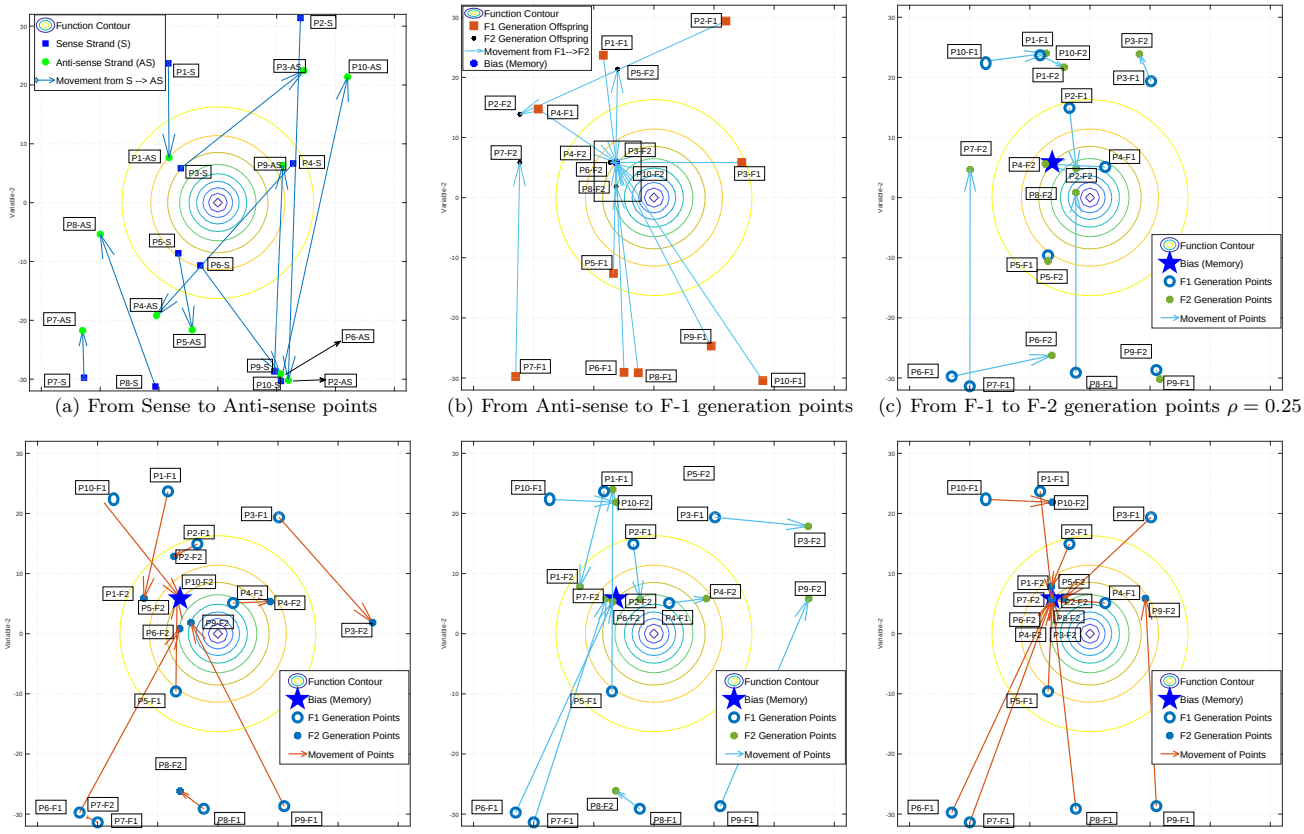
4 Movement of points

This section presents the role of different operators for movement of the points in hyperspace as a result of evolution. Flipper operation provides a stochastic search strategy which eliminates the effect of biases and prevents the optimizer from premature convergence or trapping at local minima. Effect of this operation can be seen in Fig. 11(a), where flipping the SS of DNA results in AS which spreads the points represented by S far in the search space. This operation results in an unbiased search strategy, which is very important for multi-modal problems as the global search strategy. It prevents the search to be trapped at local extremes. Distracted search from the above process is again aligned in $F1$ and $F2$ generation offspring. Due to cross-breeding $F1$ generation offspring explores the hyperspace surrounded by the two parents. As a result of F1 generation operation, the movement of points can be observed in Fig. 11(b). We can see that the new offspring are produced between two parents or the region specified by them. It shows that F1 generation offspring are influenced by their parents only and not influenced by ancestor characteristics. Here, it is worth to notice that the number of multiple offspring productions is a control variable which may vary based on the complexity of the problem to be solved.

Although self-breeding use heredity memory based on the Mendelian probability, this is the second operation, which makes the METO different from the GA procedures, where GA does not utilize heredity memory. However, few papers suggest the GA with implicit memory [58] to save the chromosome efficiently in the computer memory. This can be used with METO as well for improving computational power for large variables. Self-breeding operation pulls back the points towards the region of interest in search space. In the successive generation, heredity memory provides a biasing mechanism to produce the F2 generation offspring as a neighbor of already found best solution. This is the pull mechanism which attracts the points towards pseudo best solution acquired by this organism. The quantity of recessive heredity to transmit in $F2$ generation offspring depends on Mendelian probability ρ, which ranges from τ^L to τ^U. This provides local exploitation of the neighbor points of the pseudo-global solution, which can be seen in Fig. 11(c)–(f) for different values of ρ. We can observe that for higher ρ most of the points converge to the pseudo best point, where low value provides random location selection strategy. Thus the optimal selection of ρ can make the proposed optimizer better. This mechanism is backed by the Epi-mutation, which is in the result of the organism's survival instincts.

5 Experimental Evaluation: Benchmarking METO

In this section, experimental evaluation has been portraited to benchmark the METO. For this, we utilize some complex test functions which belong to the different categories of complexity. Details of experimental benchmark test functions are given in many publications, books, and online [37–39]. These benchmark functions belong to the different categories such as continuous, discrete, analytical, non-analytical,



(a) From Sense to Anti-sense points (b) From Anti-sense to F-1 generation points (c) From F-1 to F-2 generation points $\rho = 0.25$

(d) From F-1 to F-2 generation points $\rho = 0.50$ (e) From F-1 to F-2 generation points $\rho = 0.75$ (f) From F-1 to F-2 generation points $\rho = 0.95$

Fig. 11: Movement of points in an evolution

separable, non-separable, rotated-shifted, composite, and hybrid functions as in Table - 3 and 4. Simulation outcomes and statistical results are collected over 100 independent runs on the 30 and 100 variables problems. Simulation results also assist us in setting the limitations of METO algorithm and its parameters to get better performance. Functions F1–F20, F81 are multi-modal with many extremes test functions. F21 is deceptive function, F21–F23, and F70 are integrated functions, F19 is with many global solutions, F58–F67, F83 are Noisy functions, F68 is a constrained function, F34-F37 is rotated-shifted functions, F39–40 are the non-continuous functions, F41–F49 are the Hybrid and composite functions as described in [38].

According to the literature, many local minima functions with single global optima are hard to solve in more than twenty dimensions, where the present algorithms show some limitations. It gives motivation to the researchers for designing a better optimizer. Following the motivation, here, we focus on solving the above class of functions regarding improving the consistency with better results.

We simulate the thirty and hundred variables bench-marking test problems with twelve different prominent class of evolutionary and swarm optimizers, as shown in Table - 2. Although we have tested it with more algorithms and on more bench-mark functions, which we can provide on the demand of readers. In the limited version of the manuscript, we present some useful results to support the METO's outperformance over other algorithms. The parameters of all the optimizer are also given in Table-2. Optimal parameters of all optimizer are adopted from their corresponding papers. [19–24, 45–50] Each optimizer had a population size of 100 and run for 100000 function evaluations. 100 individual runs of each optimizer are carried out on each benchmark function to get comparative performance. Based on the output in 100 runs, three statistical measures are calculated to show the efficiency of all above optimizers. First one is the average μ of best values, which is the sum of all the final values divided by the number of runs. Second is the standard deviation σ of the achieved solution in all iterations to show the spread of the obtained results. The last attribute gives the robustness and consistency $\mathscr{C}$ of the algorithms, which is in percentage and represents the number of times the optimizer provides solution below a particular threshold value, here we selected mean of the METO is as the threshold for all optimizers. Moreover best B and worst $\mathscr{W}$ value highlights the best and worst possible performance of all optimizer. For comparison, the METO has the following parameters configuration, i.e., number of species equal to 2 with 50 number of individ-

Table 3: Test Benchmark Functions

.	Test Functions	Equations	Range
F1	Inverted Cosine Wave	$f(X) = -\sum_{i=1}^{n-1} e^{\left(\frac{-\left(x_i^2 + x_{i+1}^2 + 0.5 x_i x_{i+1}\right)}{8}\right)} \cos\left(4 \times \sqrt{x_i^2 + x_{i+1}^2 + 0.5 x_i x_{i+1}}\right)$	± 5
F2	Rastrigin	$f(x,y) = 10n + \sum_{i=1}^{n}(x_i^2 - 10cos(2\pi x_i))$	±-5.125
F3	Gen. Schwefel 226	$f(x) = -\sum_{i=1}^{n}\left[x_i \sin\left(\sqrt{\lvert x_i\rvert}\right)\right]$	± 500
F4	Wavy	$f(x) = \frac{1}{n}\sum_{i=1}^{n} 1 - \cos(10x_i)\, e^{-\frac{1}{2}x_i^2}$	± \pi
F5	Dropwave	$f(X) = -\sum_{i=1}^{n-1} \frac{1+\cos\left(12\sqrt{x_i^2 + x_{i+1}^2}\right)}{\frac{1}{2}\left(x_i^2 + x_{i+1}^2\right) + 2}$	± 5.125
F6	LangMann	$f(X) = \sum_{i=1}^{m} c_i \exp\left(-\frac{1}{\pi}\sum_{j=1}^{n}\left(x_j - A_{ij}\right)^2\right) cos\left(\pi \sum_{j=1}^{n}\left(x_j - A_{ij}\right)^2\right)$	[0, 10]
F7	Luniacek Bi Rastrigin	$f(X) = \min\left[\sum_{i=1}^{n}\left(x_i - \mu_1\right)^2,\ d \cdot n + s \cdot \sum_{i=1}^{n}\left(x_i - \mu_2\right)^2\right] + 10\sum_{i=1}^{n}\left\{1 - \cos\left[2\pi\left(x_i - \mu_1\right)\right]\right\}$ $\mu_1 = 2.5, \quad \mu_2 = -\sqrt{\frac{\mu_1^2 - d}{s}}$ $d \in \{1,2,3,4\}$ and s is any value between 0.2 and 1.4 as described in [1***]	± 5.125
F8	Suharev-Zilinskas	$f(\mathbf{x}) = \sum_{i=1}^{n}\sum_{j=1}^{5} j sin((j+1)x_i + j)$	± 10
F9	Shubert4	$f(\mathbf{x}) = \sum_{i=1}^{n}\sum_{j=1}^{5} j cos((j+1)x_i + j)$	± 10
F10	Ext. Bird	$f(x,y) = \sum_{i=1}^{n-1}\left(sin(x_i) e^{(1-cos(x_{i+1}))^2} + cos(x_{i+1}) e^{(1-sin(x_i))^2} + (x_i - x_{i+1})^2\right)$	± 10
F11	Periodic	$f(\mathbf{x}) = f(x_1 \ldots x_n) = 1 + \sum_{i=1}^{n} sin^2(x_i) - 0.1 e^{(\sum_{i=1}^{n} x_i^2)}$	[-5, 10]
F12	Ext. Gramacy Lee	$f(X) = \sum_{i=1}^{n}\left(\frac{\sin(10\pi x_i)}{2x_i} + (x_i - 1)^4\right)$	[-0.5, 2.5]
F13	Schaffer6	$f(X) = 0.5 + \sum_{i=1}^{n-1}\left(\frac{\sin^2\left(\sqrt{x_i^2 + x_{i+1}^2}\right) - 0.5}{\left[1 + 0.001\left(x_i^2 + x_{i+1}^2\right)\right]^2}\right)$	± 100
F14	Schaffer2	$f(X) = 0.5 + \sum_{i=1}^{n-1}\left(\frac{\sin^2\left(x_i^2 - x_{i+1}^2\right) - 0.5}{\left[1 + 0.001\left(x_i^2 + x_{i+1}^2\right)\right]^2}\right)$	± 100
F15	Schaffer4	$f(X) = 0.5 + \sum_{i=1}^{n-1}\left(\frac{\cos\left(\sin\left(\lvert x_i^2 - x_{i+1}^2\rvert\right)\right) - 0.5}{\left[1 + 0.001\left(x_i^2 + x_{i+1}^2\right)\right]^2}\right)$	± 100
F16	Deb01	$f(X) = -\frac{1}{n}\sum_{i=1}^{n} \sin^6(5\pi x_i)$	[-1, 1]
F17	Eggholder	$f(X) = \sum_{i=1}^{n-1}\left[-x_i \sin\left(\sqrt{\lvert x_i - x_{i+1} - 47\rvert}\right) - (x_{i+1} + 47)\sin\left(\sqrt{\lvert 0.5x_i + x_{i+1} + 47\rvert}\right)\right]$	± 512
F18	Deceptive	$f(X) = \left[\frac{1}{n}\sum_{i=1}^{n} g_i(x_i)\right]^{\beta}$ $g_i(x_i) = \begin{cases} -\frac{x_i}{\alpha_i} + \frac{4}{5} & \text{if } 0 \le x_i < \frac{4}{5}\alpha_i \\ \frac{5x_i}{\alpha_i} - 4 & \text{if } \frac{4}{5}\alpha_i \le x_i < \alpha_i \\ \frac{5(x_i - \alpha_i)}{\alpha_i - 1} + 1 & \text{if } \alpha_i \le x_i < \frac{1 + 4\alpha_i}{5} \\ \frac{x_i - 1}{1 - \alpha_i} + \frac{4}{5} & \text{otherwise} \end{cases}$ \\ β is a non-linearity factor. It is considered equal to 1 in [3] $0 < \alpha_i < 1$	[0, 1]
F19	Michalewicz	$f(X) = \sum_{j=1}^{n} \sin(x_j)\left[\sin\left(\frac{jx_j^2}{\pi}\right)\right]^{2m}$ m=10	[0,π]
F20	LorF2	$f(X) = -\prod_{i=1}^{n} \sin^k(l_1 \pi x_i + l_2) \cdot e^{-l_3\left(\frac{x_i - l_4}{l_5}\right)^2}$ $k = 6, \quad l_1 = 5.1, \quad l_2 = 0.5, \quad l_3 = 4ln(2), \quad l_4 = 0.066832364099628, \quad l_5 = 0.64$	[0, 1]
F21	Deformed Schaffer2	$f(X) = 0.5 + \sum_{i=1}^{n-1}\left(\frac{\sin^2\left(x_i^2 - x_{i+1}^2\right) - 0.5}{\left[1 + 0.01\left(x_i^2 + x_{i+1}^2\right)\right]^2}\right)$	± 100
F22	Keane Bump Constraints	$f(x) = -\lvert\frac{\sum_{i=1}^{n}\cos^4(x_i) - 2\prod_{i=1}^{n}\cos^2(x_i)}{\left(\sum_{i=1}^{n} i x_i^2\right)^{0.5}}\rvert$ subject to: $g_1(x) = 0.75 - \prod_{i=1}^{n} x_i < 0$, and $g_2(x) = \sum_{i=1}^{n} x_i - 7.5n < 0$	[0, 10]

Table 4: Test Benchmark Functions

.	Test Functions	Equations	Range
F23	IntegratedFunc1	f(x) = F4(x) + F5(x)	± \pi
F24	IntegratedFunc2	$f(x) = \text{schaffer6} + 0.5 + \sum_{1=1}^{n-1} \frac{\sin(x_i^2 - x_{i+1}^2)^2 - 0.5}{1+0.01(x_i^2 + x_{i+1}^2)^2}$	± 10
F25	IntegratedFunc3	$f(x) = \sum_{i=1}^{n} \lceil \lvert x_i + 0.5 \rvert \rceil^2$ + 30*F1(x);	± 5
F26	Noisy Schaffer2	Schaffer2 + gaussianNoise[μ, σ]	± 100
F27	Eggholder Noisy	Eggholder + gaussianNoise[μ, σ]	± 512
F28	LangMann Noisy	LangMann + gaussianNoise[μ, σ]	[0, 10]
F29	Noisy Hybrid Composition Function 1	Hybrid function 1 + gaussianNoise[μ, σ] See reference CEC 2005 Benchmark functions [[38]]	± 5
F30	Noisy Rotated Hybrid Composition Function 3	Hybrid_rot_func3 + gaussianNoise[μ, σ] See reference CEC 2005 Benchmark functions [[38]]	± 5
F31	Expanded Extended Griewank's plus Rosenbrock	$f_1(X) = 100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2$ $f(X) = \sum_{i=1}^{n-1} \left(1 + \frac{f_1(X)^2}{4000} - \cos(f_1(X))\right)$	[-3, 1]
F32	Sin envelope sin wave	$f(X) = \sum_{i=1}^{n-1} \left(\frac{\sin(\sqrt{(x_i^2 + x_{i+1}^2)})^2 - 0.5}{(1+0.01*(x_i^2 + x_{i+1}^2))^2} + 0.5 \right)$	± 100
F33	Noisy Sinusoidal	$f(X) = -[A\prod_{i=1}^{n} \sin(x_i - Z) + \prod_{i=1}^{n} \sin(B(x_i - Z))] + gaussianNoise$[\mu,\sigma] $A = 2.5, B = 5, Z = 30$	± 100
F34	Ackley Rot	Please see reference [[38]]	± 32
F35	Rastrigin Rot	Please see reference [[38]]	± 5.125
F36	ScafferF6 Rot	Please see reference [[38]]	± 100
F37	Rastrigin Noncont	Rastrigin function with modified input variables as: $x_i = (\lvert x_i \rvert < 0.5)x_i + (\lvert x \rvert \geq 0.5)\frac{[x_i.*2]}{2})$	± 5.125
F38	fE_ScafferF6_noncont	Schaffer no 6 function with modified input variables as: $x_i = (\lvert x_i \rvert < 0.5)x_i + (\lvert x \rvert \geq 0.5)\frac{[x_i.*2]}{2})$	± 100
F39	Hybrid Composition Function 1	$f_bias = 120$, two times composition of 5 functions See reference [[38]] Rastrigin, Weierstrass, Griewank, Ackley, Sphere	± 5
F40	Hybrid Rotated Composition Function 1	Hybrid Composition Function 1 is Rotated by M matrix, See reference [[38]]	± 5
F41	Noisy Hybrid Rotated Composition Function 1	Hybrid Rotated Composition Function + gaussianNoise[μ, σ]	± 5
F42	Hybrid Rotated Composition Function 3	$f_bias = 360$, two times composition of 5 functions See reference [[38]] Scaffer6, Rastrigin, EF8F2, Weierstrass, Griewank	± 5
F43	Non-Continuous Hybrid Rotated Composition Function 3	$f_bias = 360$, x variable is modified for hybrid Rotated Composite Function 3 as $x = \lvert x - o \rvert < 0.5) * x + (\lvert x - o \rvert \geq 0.5) * \frac{[x*2]}{2})$ See reference [[38]]	± 5

uals in each species, ρ^L and ρ^U are respectively 0.9 to 0.97. The cross-breeding and self-breeding rates are respectively $\max(0.3, rand)$ and $\max(0.5, rand)$. For producing the offspring, always elites parents are taken from the population. Good selection of the parameters δ_{N_b}, τ^L, and τ^U can improve the computation power of the algorithm. Kruskal Wallis non-parametric H–test is used to analyze the results [56]. This test is an extension of the Wilcoxon rank sum test to more than two groups, where we have distributions of 13 optimizers (groups). It is also a version of classical one-way ANOVA and compares the medians of the distributions to differentiate them. It gives an idea that the results of two

Table 5: Results of 30 variables test functions

		METO	BHGA	BBO	IWO	DE	CMAES	SFLA	FA	TLBO	CUCKOO	BA	GSA	SLPSO
F1	μ	**-23.6**	-22.1	-18.9	-14.9	-17.1	-6.2	-15.8	-16.2	-20.6	-12.2	-11.7	-21.7	-15.9
	σ	**1.2**	1.1	1.6	1.5	0.6	0.7	1.9	1.2	1.5	0.5	2.0	0.8	1.7
	$\mathbb{B}$	**-26.0**	-25.3	-21.2	-18.1	-18.6	-8.0	-20.1	-18.8	-22.8	-13.1	-16.9	-22.9	-19.4
	$\mathscr{W}$	**-21.4**	-20.2	-12.5	-11.7	-16.1	-4.8	-12.0	-13.3	-14.9	-11.2	-7.1	-19.8	-12.5
	$\mathscr{C}$	**46%**	12%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
F2	μ	**6.1**	34.3	21.8	45.5	60.0	152.6	66.2	74.5	7.6	110.7	107.1	6.5	15.3
	σ	**2.2**	8.7	6.3	9.5	4.9	37.7	18.5	12.2	3.4	9.7	58.3	2.2	4.4
	$\mathbb{B}$	**1.0**	16.7	13.1	20.7	45.4	4.0	24.9	52.6	0.0	89.7	0.0	1.0	5.0
	$\mathscr{W}$	**11.4**	62.5	44.9	65.4	71.4	173.6	117.4	104.6	16.9	130.3	254.5	11.9	23.9
	$\mathscr{C}$	**54%**	0%	0%	0%	0%	4%	0%	0%	32%	0%	6%	46%	2%
F3	μ	**24.8**	799.5	3185.0	5264.2	118.6	OB	4298.3	3657.6	5223.7	4131.6	5227.3	9657.2	1515.0
	σ	**37.2**	297.0	629.5	595.0	235.2	OB	1414.8	537.5	1321.5	167.5	1719.8	366.8	368.6
	$\mathbb{B}$	**1.0**	204.3	2153.0	4167.0	0.0	OB	2546.5	2445.8	2546.6	3820.7	1.2	8865.3	829.1
	$\mathscr{W}$	**166.5**	2083.5	4797.8	7287.4	1245.0	OB	8280.9	4675.6	7970.2	4452.0	6913.4	10270.2	2309.6
	$\mathscr{C}$	**60%**	0%	0%	0%	48%	0%	0%	0%	0%	0%	2%	0%	0%
F4	μ	**1.5E-2**	1.2E-1	2.6E-1	4.0E-1	2.2E-1	7.3E-1	3.1E-1	4.1E-1	3.2E-1	4.5E-1	4.8E-1	9.5E-2	5.7E-1
	σ	**9.5E-3**	2.4E-2	7.7E-2	4.8E-2	1.8E-2	2.1E-2	5.4E-2	4.6E-2	1.1E-1	2.5E-2	8.5E-2	2.3E-2	7.3E-2
	$\mathbb{B}$	**5.6E-12**	6.6E-2	1.1E-1	2.4E-1	1.8E-1	6.8E-1	1.8E-1	3.2E-1	9.3E-2	3.8E-1	2.8E-1	4.1E-2	2.7E-1
	$\mathscr{W}$	**4.4E-2**	1.7E-1	4.5E-1	4.9E-1	2.6E-1	7.8E-1	4.3E-1	5.2E-1	4.8E-1	5.0E-1	6.4E-1	1.5E-1	6.7E-1
	$\mathscr{C}$	**52%**	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
F5	μ	**-28.4**	-25.2	-26.7	-22.1	-23.9	-21.2	-21.7	-20.8	-26.8	-15.0	-19.2	-26.9	-27.1
	σ	**0.4**	0.8	0.4	1.7	0.3	2.1	1.6	0.9	0.6	0.9	2.7	0.3	0.2
	$\mathbb{B}$	**-28.9**	-26.7	-27.6	-24.8	-24.7	-27.2	-25.1	-22.9	-27.5	-17.1	-25.9	-27.4	-27.5
	$\mathscr{W}$	**-26.9**	-23.3	-25.7	-18.1	-23.2	-19.6	-18.0	-19.2	-24.3	-13.4	-14.1	-25.9	-26.6
	$\mathscr{C}$	**60%**	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
F6	μ	**-229.8**	-213.2	-161.6	-150.7	-162.2	-93.9	-171.2	-170.5	-201.1	-108.8	-107.7	-50.5	-195.6
	σ	**11.5**	11.0	20.1	14.2	7.9	44.4	25.0	11.2	15.1	4.7	19.7	8.8	16.6
	$\mathbb{B}$	**-250.1**	-236.8	-203.4	-175.7	-180.5	-218.3	-224.8	-192.6	-233.8	-124.6	-165.9	-72.3	-233.4
	$\mathscr{W}$	**-197.3**	-183.5	-118.6	-116.7	-147.7	-53.7	-81.1	-142.4	-156.5	-99.3	-72.7	-36.7	-150.0
	$\mathscr{C}$	**52%**	4%	0%	0%	0%	0%	0%	0%	2%	0%	0%	0%	2%
F7	μ	**33.6**	56.2	57.4	56.0	114.9	171.7	94.8	93.3	62.5	133.0	185.7	54.0	41.6
	σ	**13.1**	14.7	17.2	17.0	7.7	34.2	25.9	15.8	20.2	16.2	48.7	10.9	12.7
	$\mathbb{B}$	**7.0**	18.8	24.0	23.5	96.6	5.0	44.8	62.0	19.9	93.1	99.9	23.9	7.0
	$\mathscr{W}$	**49.2**	75.6	109.7	93.1	129.2	201.9	155.1	123.7	103.8	166.0	296.4	70.5	59.6
	$\mathscr{C}$	**26%**	14%	14%	12%	0%	2%	0%	0%	6%	0%	0%	12%	16%
F8	μ	**-443.8**	-415.6	-408.0	-300.6	-312.8	-134.1	-315.2	-230.4	-152.6	-238.4	-207.1	-245.9	-386.8
	σ	**1.1**	12.5	16.3	16.7	7.9	10.9	33.2	21.9	9.9	8.0	24.4	30.1	40.7
	$\mathbb{B}$	**-444.9**	-438.2	-441.6	-340.7	-330.3	-182.1	-393.9	-285.2	-175.0	-257.4	-266.2	-324.3	-433.4
	$\mathscr{W}$	**-439.9**	-382.8	-375.8	-257.2	-295.3	-119.5	-228.6	-174.1	-137.8	-216.5	-159.3	-192.1	-185.9
	$\mathscr{C}$	**64%**	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
F9	μ	**-385.2**	-359.9	-353.5	-274.6	-271.1	-124.2	-271.9	-191.1	-131.9	-216.6	-195.7	-212.7	-269.7
	σ	**1.0**	10.3	12.3	16.7	5.5	8.1	24.0	22.5	10.0	7.3	26.2	20.7	53.1
	$\mathbb{B}$	**-386.0**	-377.1	-379.1	-309.3	-282.9	-142.3	-332.6	-238.1	-174.2	-238.8	-252.6	-268.5	-348.0
	$\mathscr{W}$	**-381.4**	-336.9	-328.3	-238.9	-261.5	-113.3	-210.8	-132.6	-117.4	-197.7	-119.4	-173.8	-136.8
	$\mathscr{C}$	**84%**	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
F10	μ	**-1.3E+3**	-1.2E+3	-1.3E+3	-1.1E+3	-7.6E+2	-1.4E+3	-1.2E+3	-1.1E+3	-1.2E+3	-8.3E+2	-7.6E+2	-1.2E+3	-1.3E+3
	σ	**73.7**	71.1	90.6	111.9	73.1	54.8	96.3	77.1	111.6	37.9	123.8	250.1	68.0
	$\mathbb{B}$	**-1.5E+3**	-1.3E+3	-1.4E+3	-1.3E+3	-9.5E+2	-1.5E+3	-1.4E+3	-1.3E+3	-1.4E+3	-9.1E+2	-1.2E+3	-1.5E+3	-1.4E+3
	$\mathscr{W}$	**-1.1E+3**	-1.0E+3	-1.0E+3	-6.6E+2	-5.8E+2	-1.3E+3	-9.3E+2	-8.9E+2	-9.7E+2	-7.4E+2	-4.4E+2	-4.9E+2	-1.1E+3
	$\mathscr{C}$	**50%**	0%	16%	0%	0%	90%	6%	0%	10%	0%	0%	28%	32%
F11	μ	1.0	1.1	1.0	1.1	2.6	7.8	1.0	1.3	2.2	1.2	1.1	1.0	1.0
	σ	**0.0**	0.0	0.0	0.0	0.2	0.4	0.0	0.1	1.4	0.0	0.1	0.0	0.0
	$\mathbb{B}$	**1.0**	1.0	1.0	1.0	2.2	6.8	1.0	1.1	1.0	1.1	1.0	1.0	1.0
	$\mathscr{W}$	**1.0**	1.2	1.0	1.1	3.0	8.6	1.0	1.7	5.9	1.2	1.7	1.0	1.0
	$\mathscr{C}$	**64%**	0%	100%	0%	0%	0%	100%	0%	6%	0%	18%	100%	100%

Table 6: Results of 30 variables test functions

		METO	BHGA	BBO	IWO	DE	CMAES	SFLA	FA	TLBO	CUCKOO	BA	GSA	SLPSO
F12	μ	**-84.8**	-59.6	-38.5	-26.5	-59.0	-13.3	-38.2	-23.3	-21.4	-27.0	-24.4	-12.1	-22.3
	σ	**5.5**	4.1	4.0	5.4	3.1	1.0	6.7	2.9	3.3	1.5	5.2	0.9	6.1
	$\mathbb{B}$	**-99.6**	-68.9	-49.7	-39.5	-65.9	-16.3	-54.2	-31.7	-30.1	-30.7	-36.7	-13.3	-49.0
	$\mathscr{W}$	**-59.6**	-49.3	-31.2	-17.4	-51.6	-11.0	-24.7	-16.7	-15.9	-24.2	-14.7	-9.6	-12.8
	$\mathscr{C}$	**56%**	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
F13	μ	**-12.5**	-10.5	-7.8	-1.8	-7.2	-1.7	-5.7	-3.1	-5.0	-2.5	-4.2	-11.5	-2.3
	σ	**0.5**	0.9	1.0	0.5	0.3	0.3	1.5	0.4	0.6	0.3	1.5	0.6	0.2
	$\mathbb{B}$	**-13.4**	-12.2	-10.0	-3.3	-7.9	-2.3	-8.6	-4.6	-6.7	-3.4	-7.7	-12.7	-2.9
	$\mathscr{W}$	**-11.1**	-8.1	-5.4	-0.9	-6.4	-1.2	-1.8	-2.5	-4.1	-2.1	-1.9	-10.0	-1.6
	$\mathscr{C}$	**50%**	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	4%	0%
F14	μ	**-12.3**	-8.6	-7.4	-1.5	-7.2	-1.7	-3.1	-3.0	-4.7	-2.0	-4.8	-11.8	-2.3
	σ	**0.6**	0.8	1.1	0.4	0.3	0.3	0.7	0.3	0.4	0.2	1.2	0.5	0.2
	$\mathbb{B}$	**-13.4**	-10.4	-10.2	-2.5	-8.0	-2.5	-6.0	-4.0	-5.7	-2.7	-8.1	-13.0	-2.8
	$\mathscr{W}$	**-11.0**	-7.0	-4.8	-0.7	-6.6	-1.2	-2.0	-2.3	-4.0	-1.6	-2.4	-10.1	-1.8
	$\mathscr{C}$	**52%**	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	18%	0%
F15	μ	**-4.0**	-2.3	-1.7	0.1	-1.5	0.2	-0.2	-0.2	-0.1	-0.1	-0.7	-0.1	0.1
	σ	**0.3**	0.4	0.3	0.1	0.1	0.1	0.4	0.1	0.1	0.0	0.3	0.2	0.0
	$\mathbb{B}$	**-4.8**	-3.1	-2.3	-0.3	-1.8	0.0	-1.8	-0.4	-0.3	-0.2	-1.5	-0.5	-0.1
	$\mathscr{W}$	**-3.3**	-1.7	-0.9	0.3	-1.3	0.3	0.1	-0.1	0.0	-0.1	-0.3	0.1	0.1
	$\mathscr{C}$	**54%**	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
F16	μ	-1.00	-0.99	-1.00	-0.81	-0.88	-0.61	-1.00	-0.88	-0.66	-0.95	-0.94	-1.00	-0.97
	σ	0.00	0.01	0.00	0.12	0.01	0.02	0.00	0.02	0.03	0.01	0.05	0.00	0.04
	$\mathbb{B}$	-1.00	-1.00	-1.00	-1.00	-0.91	-0.66	-1.00	-0.92	-0.77	-0.96	-1.00	-1.00	-1.00
	$\mathscr{W}$	-1.00	-0.98	-1.00	-0.55	-0.85	-0.58	-1.00	-0.82	-0.62	-0.93	-0.83	-1.00	-0.86
	$\mathscr{C}$	58%	0%	20%	0%	0%	0%	100%	0%	0%	0%	4%	100%	32%
F17	μ	**-2.2E+4**	-1.8E+4	-1.4E+4	-1.2E+4	-1.4E+4	OB	-1.1E+4	-1.5E+4	-1.0E+4	-1.4E+4	-1.3E+4	-4.4E+3	-1.8E+4
	σ	**1.6E+3**	1.3E+3	1.1E+3	1.3E+3	5.3E+2	OB	2.5E+3	1.2E+3	1.8E+3	3.4E+2	2.5E+3	6.0E+2	1.1E+3
	$\mathbb{B}$	**-2.6E+4**	-2.2E+4	-1.7E+4	-1.4E+4	-1.6E+4	OB	-1.5E+4	-1.8E+4	-1.8E+4	-1.5E+4	-2.0E+4	-6.3E+3	-2.0E+4
	$\mathscr{W}$	**-1.8E+4**	-1.6E+4	-1.2E+4	-9.2E+3	-1.3E+4	OB	-5.5E+3	-1.3E+4	-8.3E+3	-1.3E+4	-9.6E+3	-3.3E+3	-1.5E+4
	$\mathscr{C}$	**54%**	2%	0%	0%	0%	OB	0%	0%	0%	0%	0%	0%	0%
F18	μ	-0.96	-0.83	-0.82	-0.63	-0.98	OB	-0.49	-0.70	-0.68	-0.63	-0.65	-0.77	-0.69
	σ	0.02	0.03	0.03	0.03	0.01	OB	0.20	0.02	0.02	0.01	0.01	0.02	0.02
	$\mathbb{B}$	-0.99	-0.88	-0.88	-0.70	-1.00	OB	-0.83	-0.75	-0.76	-0.66	-0.66	-0.80	-0.77
	$\mathscr{W}$	-0.92	-0.79	-0.75	-0.58	-0.96	OB	-0.32	-0.64	-0.61	-0.61	-0.63	-0.72	-0.65
	$\mathscr{C}$	56%	0%	0%	0%	84%	OB	0%	0%	0%	0%	0%	0%	0%
F19	μ	**-29.0**	-26.6	-26.0	-20.1	-23.6	-12.9	-25.1	-16.0	-26.4	-16.7	-15.7	-28.8	-16.4
	σ	**0.2**	0.7	0.9	2.6	0.4	0.6	0.9	0.9	3.2	0.5	1.9	0.3	1.3
	$\mathbb{B}$	**-29.5**	-27.8	-27.5	-24.7	-24.6	-14.8	-27.4	-17.9	-28.7	-17.8	-20.6	-29.2	-19.6
	$\mathscr{W}$	**-28.4**	-25.0	-23.8	-14.2	-22.7	-11.9	-22.8	-14.1	-16.2	-15.6	-13.1	-27.3	-14.1
	$\mathscr{C}$	**52%**	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	26%	0%
F20	μ	**-0.72**	-0.03	0.00	0.00	0.00	0.00	0.00	0.00	-0.09	0.00	0.00	0.00	-0.03
	σ	**0.15**	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.08	0.00	0.00	0.00	0.04
	$\mathbb{B}$	**-0.93**	-0.33	0.00	0.00	-0.01	0.00	0.00	0.00	-0.36	0.00	0.00	0.00	-0.16
	$\mathscr{W}$	**-0.32**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.01	0.00	0.00	0.00	0.00
	$\mathscr{C}$	**40%**	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
F21	μ	**-12.5**	-7.9	-6.9	-1.1	-6.5	-0.7	-11.6	-2.4	-2.9	-1.4	-12.6	-5.9	-5.4
	σ	**0.8**	0.9	1.1	0.4	0.4	0.2	4.3	0.3	0.5	0.1	3.0	1.5	1.5
	$\mathbb{B}$	**-13.8**	-9.9	-9.3	-2.0	-7.3	-1.3	-14.0	-3.0	-4.7	-1.8	-14.0	-9.1	-8.0
	$\mathscr{W}$	**-10.5**	-6.1	-4.5	-0.5	-5.9	-0.3	-0.9	-1.7	-1.8	-1.1	-3.8	-3.3	-1.3
	$\mathscr{C}$	**52%**	0%	0%	0%	0%	0%	70%	0%	0%	0%	78%	0%	0%
F22	μ	**-0.80**	-0.76	-0.48	-0.47	-0.44	-0.46	-0.63	-0.45	-0.44	-0.44	-0.44	-0.42	-0.51
	σ	**0.01**	0.02	0.04	0.09	0.00	0.02	0.12	0.03	0.00	0.00	0.00	0.01	0.13
	$\mathbb{B}$	**-0.81**	-0.80	-0.57	-0.75	-0.44	-0.56	-0.79	-0.65	-0.44	-0.46	-0.44	-0.44	-0.80
	$\mathscr{W}$	**-0.77**	-0.69	-0.41	-0.33	-0.44	-0.44	-0.34	-0.44	-0.44	-0.44	-0.44	-0.38	-0.44
	$\mathscr{C}$	**56%**	2%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	2%

Table 7: Results of 30 variables test functions

		METO	BHGA	BBO	IWO	DE	CMAES	SFLA	FA	TLBO	CUCKOO	BA	GSA	SLPSO
F23	μ	**-28.0**	-24.8	-25.3	-23.3	-23.0	-20.5	-21.8	-21.7	-26.1	-17.4	-19.0	-26.6	-26.7
	σ	**0.7**	0.6	0.8	1.0	0.3	2.5	1.1	0.9	0.6	0.7	2.5	0.4	0.4
	$\mathbb{B}$	**-29.0**	-26.4	-27.4	-25.5	-24.1	-26.9	-24.6	-23.8	-27.4	-18.9	-24.5	-27.4	-27.2
	$\mathscr{W}$	**-26.2**	-23.2	-22.9	-21.1	-22.4	-18.7	-19.7	-19.1	-24.9	-16.1	-13.4	-25.3	-25.5
	$\mathscr{C}$	**60%**	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
F24	μ	-26.1	-21.0	-19.0	-17.1	-14.2	-6.1	-18.2	-14.4	-11.8	-13.0	-12.9	-25.5	-19.0
	σ	1.2	1.4	1.4	1.2	0.6	0.7	1.3	1.2	1.5	0.5	1.6	0.8	1.0
	$\mathbb{B}$	-28.0	-24.2	-22.0	-19.1	-15.9	-8.9	-21.0	-17.0	-16.9	-14.2	-17.1	-27.1	-22.4
	$\mathscr{W}$	-23.1	-17.9	-16.1	-14.4	-12.8	-4.9	-15.2	-11.4	-9.4	-12.1	-9.3	-23.8	-17.0
	$\mathscr{C}$	56%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	30%	0%
F25	μ	**-694.3**	-643.0	-627.5	-539.1	-528.8	-600.7	-504.8	-531.5	-640.0	-383.5	-348.3	-652.4	-620.1
	σ	**38.6**	35.0	25.3	46.6	15.5	87.7	50.4	44.7	22.2	29.2	95.4	25.5	18.2
	$\mathbb{B}$	**-791.9**	-713.7	-674.1	-623.2	-571.9	-646.9	-594.6	-636.5	-688.0	-453.8	-589.2	-714.5	-646.3
	$\mathscr{W}$	**-619.2**	-551.7	-561.5	-427.1	-501.2	-271.4	-389.0	-431.9	-583.4	-313.6	-129.8	-605.4	-569.8
	$\mathscr{C}$	**50%**	6%	0%	0%	0%	0%	0%	0%	0%	0%	0%	8%	0%
F26	μ	-11.2	-8.2	-6.0	-2.8	-3.4	-2.1	-3.0	-3.2	-3.7	-2.7	-4.1	-13.7	-2.8
	σ	1.1	1.0	0.8	0.3	0.3	0.2	0.4	0.3	0.5	0.3	1.2	0.7	0.3
	$\mathbb{B}$	-13.0	-10.0	-8.4	-3.6	-4.1	-2.5	-4.9	-4.3	-5.2	-3.6	-7.5	-15.1	-3.7
	$\mathscr{W}$	-8.8	-6.1	-4.3	-2.3	-2.8	-1.8	-2.5	-2.8	-3.0	-2.2	-2.7	-11.8	-2.3
	$\mathscr{C}$	48%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%	0%
F27	μ	**-1.9E+4**	-1.8E+4	-1.4E+4	-1.1E+4	-1.4E+4	OB	-1.1E+4	-1.5E+4	-1.1E+4	-1.4E+4	-1.2E+4	-4.3E+3	-1.8E+4
	σ	**1.3E+3**	9.3E+2	1.3E+3	1.6E+3	4.4E+2	OB	2.4E+3	1.0E+3	1.8E+3	3.6E+2	1.9E+3	6.2E+2	1.2E+3
	$\mathbb{B}$	**-2.2E+4**	-2.0E+4	-1.7E+4	-1.5E+4	-1.5E+4	OB	-1.5E+4	-1.8E+4	-1.7E+4	-1.5E+4	-1.9E+4	-6.1E+3	-2.0E+4
	$\mathscr{W}$	**-1.7E+4**	-1.6E+4	-1.1E+4	-6.9E+3	-1.4E+4	OB	-5.6E+3	-1.3E+4	-8.1E+3	-1.3E+4	-9.5E+3	-3.2E+3	-1.5E+4
	$\mathscr{C}$	**52%**	12%	0%	0%	0%	OB	0%	0%	0%	0%	0%	0%	10%
F28	μ	**-227.4**	-215.7	-153.1	-149.4	-160.3	-82.7	-169.0	-171.1	-194.0	-106.2	-95.9	-53.0	-195.5
	σ	**8.9**	11.5	21.8	12.9	4.8	29.4	19.4	13.5	16.6	5.3	21.6	9.8	14.5
	$\mathbb{B}$	**-250.4**	-235.4	-188.8	-179.0	-170.9	-213.7	-197.3	-200.8	-224.7	-118.7	-153.0	-77.5	-225.9
	$\mathscr{W}$	**-205.2**	-190.4	-109.0	-122.3	-150.4	-54.2	-114.6	-146.1	-157.3	-94.6	-52.2	-37.9	-158.1
	$\mathscr{C}$	**48%**	18%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
F29	μ	618.9	675.1	638.2	654.1	736.9	738.7	617.9	669.1	693.6	735.9	940.8	601.1	680.5
	σ	21.4	56.7	73.0	138.2	11.6	89.9	28.0	28.6	121.5	21.9	88.7	76.4	82.4
	$\mathbb{B}$	579.8	624.0	575.2	558.6	708.2	667.5	567.1	612.8	574.5	688.9	729.4	564.3	549.3
	$\mathscr{W}$	695.6	912.1	899.6	901.3	761.8	896.0	759.1	723.5	974.5	789.0	1169.4	898.7	898.1
	$\mathscr{C}$	52%	0%	54%	74%	0%	0%	58%	6%	36%	0%	0%	94%	16%
F30	μ	1.3E+3	1.3E+3	1.3E+3	1.3E+3	1.3E+3	8.5E+2	1.3E+3	1.3E+3	1.3E+3	1.3E+3	1.3E+3	1.4E+3	1.3E+3
	σ	55.7	55.9	47.9	63.3	62.5	160.1	47.2	56.1	54.8	62.1	69.7	50.8	75.8
	$\mathbb{B}$	1.1E+3	1.1E+3	1.2E+3	1.1E+3	1.2E+3	6.0E+2	1.2E+3	1.2E+3	1.1E+3	1.1E+3	1.1E+3	1.3E+3	1.0E+3
	$\mathscr{W}$	1.5E+3	1.4E+3	1.4E+3	1.4E+3	1.4E+3	1.3E+3	1.4E+3	1.4E+3	1.4E+3	1.5E+3	1.4E+3	1.5E+3	1.4E+3
	$\mathscr{C}$	54%	52%	26%	74%	42%	100%	24%	48%	46%	52%	54%	4%	56%
F31	μ	3.0	5.9	2.7	14.4	9.2	11.7	4.5	15.5	5.6	13.3	35.4	8.5	4.2
	σ	0.6	1.5	0.4	2.2	0.7	2.7	1.2	1.7	3.5	1.3	26.8	1.2	1.5
	$\mathbb{B}$	1.8	2.3	1.8	7.5	7.3	6.8	2.3	11.5	2.0	10.7	9.6	5.7	2.4
	$\mathscr{W}$	4.5	9.5	3.5	18.2	10.5	15.0	7.9	18.7	14.4	16.6	150.0	11.4	8.9
	$\mathscr{C}$	50%	4%	78%	0%	0%	0%	6%	0%	12%	0%	0%	0%	8%
F32	μ	**1.7**	3.5	6.2	12.2	6.7	12.3	8.5	10.9	9.0	11.5	9.6	2.5	11.6
	σ	**0.6**	0.8	0.9	0.5	0.4	0.3	1.0	0.4	0.5	0.3	1.1	0.5	0.3
	$\mathbb{B}$	**0.8**	1.9	4.0	10.9	5.3	11.5	5.6	9.7	8.0	10.4	6.8	1.4	10.9
	$\mathscr{W}$	**3.0**	5.3	8.3	12.9	7.4	13.0	10.6	11.4	9.8	11.9	12.2	3.6	12.1
	$\mathscr{C}$	**48%**	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	4%	0%
F33	μ	-1.39	-1.40	-1.40	-1.38	-1.39	-1.40	-1.37	-1.39	-1.39	-1.40	-1.53	-1.38	-1.39
	σ	0.09	0.09	0.09	0.08	0.08	0.08	0.08	0.07	0.08	0.07	0.20	0.08	0.08
	$\mathbb{B}$	-1.61	-1.63	-1.63	-1.57	-1.62	-1.62	-1.60	-1.55	-1.58	-1.59	-2.29	-1.60	-1.73
	$\mathscr{W}$	-1.23	-1.22	-1.21	-1.23	-1.19	-1.27	-1.26	-1.24	-1.23	-1.27	-1.21	-1.25	-1.25
	$\mathscr{C}$	48%	46%	54%	30%	44%	46%	32%	40%	44%	44%	72%	38%	40%

Table 8: Results of 30 variables test functions

		METO	BHGA	BBO	IWO	DE	CMAES	SFLA	FA	TLBO	CUCKOO	BA	GSA	SLPSO
F34	μ	20.9	20.9	20.9	20.6	21.0	21.0	21.0	21.0	21.0	21.0	20.9	20.2	21.0
	σ	0.1	0.1	0.1	0.3	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0
	$\mathbb{B}$	20.7	20.7	20.7	20.1	20.8	20.8	20.8	20.9	20.8	20.8	20.7	20.1	20.9
	$\mathscr{W}$	21.0	21.0	21.0	21.1	21.1	21.1	21.1	21.1	21.1	21.1	21.1	20.4	21.1
	$\mathscr{C}$	46%	26%	66%	60%	4%	18%	4%	4%	6%	6%	42%	100%	4%
F35	μ	**9.5**	35.5	29.9	42.7	51.9	153.7	78.3	80.3	77.3	132.7	227.9	42.5	16.0
	σ	**2.6**	7.6	8.6	8.2	4.6	38.1	20.0	13.1	16.6	15.2	47.5	6.3	5.8
	$\mathbb{B}$	**3.7**	21.3	17.4	22.8	42.7	6.0	48.8	55.1	40.8	95.5	138.0	26.9	5.0
	$\mathscr{W}$	**16.1**	50.9	63.8	63.4	63.2	179.4	142.3	105.3	106.5	169.8	374.9	55.7	30.8
	$\mathscr{C}$	**52%**	0%	0%	0%	0%	2%	0%	0%	0%	0%	0%	0%	8%
F36	μ	12.8	13.0	12.8	13.8	13.5	13.6	13.1	13.4	13.2	13.3	13.2	14.1	13.1
	σ	0.4	0.4	0.4	0.3	0.2	0.2	0.4	0.2	0.2	0.1	0.4	0.1	0.2
	$\mathbb{B}$	11.8	11.6	11.6	12.7	13.0	13.2	11.8	12.9	12.8	12.9	12.1	13.5	12.6
	$\mathscr{W}$	13.3	13.6	13.5	14.2	13.8	14.0	13.7	13.6	13.6	13.6	14.0	14.3	13.6
	$\mathscr{C}$	48%	28%	40%	2%	0%	0%	16%	0%	4%	0%	16%	0%	6%
F37	μ	**4.36**	24.84	24.40	63.69	37.33	132.01	85.76	63.49	17.97	112.50	106.34	8.20	24.39
	σ	**1.55**	4.03	3.75	15.42	2.70	11.56	20.41	15.38	9.23	15.73	58.57	2.43	4.51
	$\mathbb{B}$	**0.23**	17.01	17.01	39.07	29.82	93.07	52.00	37.46	9.16	69.24	12.00	4.00	13.00
	$\mathscr{W}$	**8.19**	35.00	34.00	95.21	43.51	153.69	137.00	94.87	61.86	141.12	229.44	14.00	34.37
	$\mathscr{C}$	**62%**	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	6%	0%
F38	μ	**2.04**	3.92	6.78	12.78	7.20	12.69	8.99	11.33	9.46	12.06	10.62	3.19	12.13
	σ	**0.61**	0.83	0.78	0.35	0.38	0.37	1.33	0.26	0.50	0.25	0.94	0.55	0.28
	$\mathbb{B}$	**0.89**	2.26	5.32	11.76	5.64	11.77	5.27	10.53	8.21	11.31	8.44	2.22	11.49
	$\mathscr{W}$	**3.67**	5.84	8.54	13.44	7.68	13.38	12.18	11.80	10.17	12.51	12.70	4.41	12.61
	$\mathscr{C}$	**54%**	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
F39	μ	586.1	712.4	726.8	713.3	489.0	720.6	834.6	591.7	794.0	811.4	1047.6	900.0	715.3
	σ	99.2	125.0	97.1	103.1	24.2	287.4	110.3	64.1	124.1	13.7	78.6	0.0	136.6
	$\mathbb{B}$	497.9	554.8	588.0	563.1	481.2	200.0	561.7	566.6	573.0	787.0	796.1	900.0	486.6
	$\mathscr{W}$	900.4	973.2	900.0	900.2	600.8	908.3	968.8	900.2	981.2	842.0	1168.1	900.0	900.0
	$\mathscr{C}$	54%	18%	0%	4%	96%	24%	2%	74%	2%	0%	0%	0%	4%
F40	μ	619.0	675.1	636.7	652.4	741.1	752.0	654.6	669.0	646.5	734.5	924.2	597.6	688.7
	σ	15.8	74.9	60.8	138.1	13.2	129.1	102.6	27.0	91.1	20.0	91.4	77.5	82.6
	$\mathbb{B}$	585.1	604.7	580.0	557.8	703.7	201.0	583.9	602.6	573.2	699.7	722.9	565.1	546.2
	$\mathscr{W}$	668.5	909.7	900.3	901.7	776.8	900.0	900.9	716.1	975.7	788.4	1085.1	900.0	900.0
	$\mathscr{C}$	50%	10%	44%	72%	0%	4%	62%	4%	56%	0%	0%	94%	14%
F41	μ	624.8	677.5	628.0	652.8	734.4	769.6	638.2	670.5	718.7	736.0	923.0	585.3	684.4
	σ	22.5	80.6	56.3	138.4	13.7	202.1	91.6	27.1	132.6	21.2	74.4	46.0	70.3
	$\mathbb{B}$	572.3	608.0	584.3	556.1	701.3	540.2	561.7	594.7	584.4	700.3	756.3	565.6	548.3
	$\mathscr{W}$	679.2	914.4	899.0	901.7	770.1	1504.4	899.9	715.2	903.9	797.8	1045.7	898.7	898.2
	$\mathscr{C}$	56%	16%	66%	72%	0%	2%	78%	6%	42%	0%	0%	98%	10%
F42	μ	1.2E+3	1.2E+3	1.2E+3	1.2E+3	1.3E+3	1.2E+3	1.2E+3	1.2E+3	1.2E+3	1.3E+3	1.2E+3	1.3E+3	1.2E+3
	σ	32.8	27.3	20.5	26.1	26.7	19.8	28.4	27.9	23.6	30.4	42.7	37.9	28.6
	$\mathbb{B}$	1.1E+3	1.2E+3	1.2E+3	1.1E+3	1.2E+3	1.2E+3	1.2E+3	1.1E+3	1.2E+3	1.2E+3	1.1E+3	1.2E+3	1.1E+3
	$\mathscr{W}$	1.3E+3	1.3E+3	1.2E+3	1.3E+3	1.3E+3	1.3E+3	1.3E+3	1.3E+3	1.3E+3	1.3E+3	1.3E+3	1.3E+3	1.3E+3
	$\mathscr{C}$	46%	48%	74%	62%	18%	64%	78%	50%	76%	16%	40%	18%	72%
F43	μ	1.3E+3	1.3E+3	1.4E+3	1.3E+3	1.3E+3	7.9E+2	1.3E+3	1.3E+3	1.3E+3	1.3E+3	1.3E+3	1.4E+3	1.3E+3
	σ	61.1	53.3	43.3	57.5	60.1	137.4	52.1	50.6	54.0	66.8	60.4	57.7	51.6
	$\mathbb{B}$	1.1E+3	1.2E+3	1.2E+3	1.2E+3	1.2E+3	5.4E+2	1.2E+3	1.2E+3	1.1E+3	1.1E+3	1.2E+3	1.3E+3	1.2E+3
	$\mathscr{W}$	1.4E+3	1.4E+3	1.4E+3	1.4E+3	1.4E+3	1.1E+3	1.4E+3	1.4E+3	1.4E+3	1.5E+3	1.4E+3	1.5E+3	1.4E+3
	$\mathscr{C}$	50%	54%	12%	58%	42%	100%	20%	48%	44%	42%	50%	4%	56%

optimizers in 100 runs are similar or different. To compute the test statistics accumulated by 100 individual runs, the Kruskal-Wallis test (KWT) uses the ranks of the distributions instead of their numeric values. This rank goes from smallest to largest values across the distribution of all optimizers. Thereafter, from the ordered data set, KWT considers the associated numeric index for the calculations. KWT utilizes "chi-square statistic" instead of F-statistic used as in the classical one-way ANOVA. The measured "p-value" reveals the significance of the calculated chi-square statistic [57]. It tests the Hypothesis that results of all optimizers are the same where the calculated results are coming from the mutually independent runs.

5.1 Observations and discussion on thirty variables problems

Associated results for thirty variables test problems are shown in Table-5 to 8 and explained below. OB in the Table associated with CMAES algorithm represents the out-of-bound solution provided by the corresponding algorithm.

F1 F1 function is best solved by the METO with $\mathbb{B} = -26.037$, $\mu = -23.6$ and median $\psi = -23.5$. Here, we can observe that approximately equal μ and ψ drives symmetrical distribution with spread $\sigma = 1.2$. To get the above mean, METO achieves high consistency which is $\mathscr{C} = 46\%$. On this function, METO's worst value $\mathscr{W}$ is the best amongst the other optimizers.

F2 On this function, METO outperforms the other optimizers, where it ends up with the best solution of $\mathbb{B} = 1$, $\mu = 6.1$, $\psi = 5.7$ and $\sigma = 2.2$. The distribution of final values in 100 individual runs is left skewed where $\psi < \mu$, which is best for getting high consistency of $\mathscr{C} = 54\%$. Also, the worst performance of METO, $\mathscr{W}$, is better than other optimizers.

F3 On Schwefel function No 226, obtained distribution from METO is left skewed due to $\psi = 1.6 < \mu = 24.8$, thus achieved consistency is the highest as $\mathscr{C} = 60\%$. Also, the best obtained solution $\mathbb{B} = 1$ is achieved by METO with minimum uncertainty, $\sigma = 37.2$. In the case of the worst solution, it is better than the other optimizers.

F4 On this function, METO also outperforms with $\mathscr{C} = 52\%$ $\psi = \mu = 1.5E^{-02}$, $\mathbb{B} = 5.6E^{-12}$ with very low uncertainty $\sigma = 9.5E^{-03}$.

F5 On Langermann function, METO give the solution with uncertainty $\sigma = 0.4$ lower than others and $\mu = -28.4$. Left skewed distribution of the solutions provides 60% consistency for getting solution below -28.4 where the other algorithms fail to achieve.

F6 Luniacek–Bi–Rastrigin function is solved best by METO with highest consistency which is more than 52% and approximately similar $\psi = -230.3$ and $\mu = -229.8$. The Worst performance METO is better than others too.

F7 On this function, METO is again best with $\mu = 33.6$ and $\psi = 40.3$, higher median than mean, which indicates that the distribution is right-skewed. Due to this, consistency is lower, but higher than other optimizers as can be seen in Table-5. The worst performance is also better than others.

F8 On this function, METO is the winner optimizer with comparatively low uncertainty of $\sigma = 1.1$ and $\psi = -444.3, \mu = -443.8$. The lower median is always preferable as indicates the better optimizer. The worst result by METO is better than others.

F9 METO is the best optimizer for this function with the normal distribution in the results, where $\psi = \mu$, as can be seen in Table-5. Moreover, METO has the least uncertainty and the highest consistency level over -385.2 value. Its worst performance is also better than others.

F10 Bird function is extended to the multiple variables, on which METO performance is comparitive to other optimizers. CMAES is giving best solution on this function.

F11 On Periodic function, four optimizer METO, BBO, GSA, and SLPSO are showing the equivalent performance, which can be seen in Table 5.

F12 Gramacy Lee function is extended to the multiple variables as shown in Tanble 3. The results of all optimizers are given in Tbale 6, where we can observe that the $(\psi = -85.5) < (\mu = 84.8)$ with better consistency $\mathscr{C} = 56\%$. METO has the best performance, also the worst performance of METO is comparatively better than others.

F13 As usual performance of METO on multimodal functions, on Schaffer6, F16 funciton, it has best response with the minimum $\mu = \psi = -12.5$, and the best worst performance. METO is able to achieve $\mathbb{B} = -13.4$ with low distribution spread and highest $\mathscr{C} = 50\%$.

F14 METO is the winner on this function in all aspects shown in Table-6. For this function, ψ is slightly lower than the μ, which is good sign for getting high consistency lower than the threshold value of 12.3.

F15 As shown in Table-6, with the equal $\psi = -4$ and $\mu = -4$, METO has the highest consistency and the best solution $\mathbb{B} = -4.8$ comparing to the others.

F16 Algorithms METO, GA, BBO, SFLA, GSA and SLPSO are giving comparatively similar with very consistent noise free solution. Deb01 function is multimodal with many minima. However, we can observe that METO outperforms on multimodal with single or very few global solutions.

F17 METO outperforms others on this function. Also its worst performance is better than others, however, the uncertainty in the distribution is high with $\sigma = 1.2e{+}3$. Also, $\psi = -1.95e{+}4$ is greater than the $\mu = -2e{+}4$.

F18 On Deceptive function, only DE competes with METO, where both algorithms are very consistent and giving a

Table 9: Kruskal Wallis ANOVA Table

	Sr	SS	df	MS	$\tilde{\chi}^2$	prob > $\tilde{\chi}^2$
F1	*Cl*	2.06E+7	1.20E+1	1.72E+6	5.84E+2	2.77E-117
	e	2.29E+6	6.37E+2	3.60E+3		
	T	2.29E+7	6.49E+2			
F2	*Cl*	1.97E+7	1.20E+1	1.64E+6	5.59E+2	5.58E-112
	e	3.17E+6	6.37E+2	4.97E+3		
	T	2.29E+7	6.49E+2			
F3	*Cl*	1.99E+7	1.20E+1	1.66E+6	5.65E+2	3.09E-113
	e	2.96E+6	6.37E+2	4.65E+3		
	T	2.29E+7	6.49E+2			
F4	*Cl*	2.07E+7	1.20E+1	1.72E+6	5.86E+2	9.34E-118
	e	2.21E+6	6.37E+2	3.48E+3		
	T	2.29E+7	6.49E+2			
F5	*Cl*	1.96E+7	1.20E+1	1.63E+6	5.56E+2	3.04E-111
	e	3.29E+6	6.37E+2	5.17E+3		
	T	2.29E+7	6.49E+2			
F6	*Cl*	1.95E+7	1.20E+1	1.63E+6	5.54E+2	7.61E-111
	e	3.36E+6	6.37E+2	5.27E+3		
	T	2.29E+7	6.49E+2			
F7	*Cl*	1.84E+7	1.20E+1	1.53E+6	5.20E+2	9.88E-104
	e	4.53E+6	6.37E+2	7.12E+3		
	T	2.29E+7	6.49E+2			
F8	*Cl*	2.15E+7	1.20E+1	1.79E+6	6.09E+2	1.00E-122
	e	1.39E+6	6.37E+2	2.19E+3		
	T	2.29E+7	6.49E+2			
F9	*Cl*	2.09E+7	1.20E+1	1.74E+6	5.92E+2	4.54E-119
	e	2.00E+6	6.37E+2	3.14E+3		
	T	2.29E+7	6.49E+2			
F10	*Cl*	1.65E+7	1.20E+1	1.37E+6	4.68E+2	1.79E-92
	e	6.40E+6	6.37E+2	1.00E+4		
	T	2.29E+7	6.49E+2			
F11	*Cl*	2.12E+7	1.20E+1	1.77E+6	6.02E+2	4.17E-121
	e	1.66E+6	6.37E+2	2.61E+3		
	T	2.29E+7	6.49E+2			
F13	*Cl*	2.15E+7	1.20E+1	1.79E+6	6.10E+2	7.32E-123
	e	1.37E+6	6.37E+2	2.15E+3		
	T	2.29E+7	6.49E+2			
F14	*Cl*	2.19E+7	1.20E+1	1.82E+6	6.20E+2	5.48E-125
	e	1.02E+6	6.37E+2	1.60E+3		
	T	2.29E+7	6.49E+2			
F15	*Cl*	2.06E+7	1.20E+1	1.72E+6	5.85E+2	1.40E-117
	e	2.24E+6	6.37E+2	3.52E+3		
	T	2.29E+7	6.49E+2			
F16	*Cl*	2.07E+7	1.20E+1	1.73E+6	5.88E+2	4.17E-118
	e	2.16E+6	6.37E+2	3.38E+3		
	T	2.29E+7	6.49E+2			
F17	*Cl*	1.90E+7	1.20E+1	1.58E+6	5.39E+2	9.20E-108
	e	3.87E+6	6.37E+2	6.07E+3		
	T	2.29E+7	6.49E+2			
F18	*Cl*	2.02E+7	1.20E+1	1.69E+6	5.74E+2	4.60E-115
	e	2.66E+6	6.37E+2	4.17E+3		
	T	2.29E+7	6.49E+2			
F19	*Cl*	2.11E+7	1.20E+1	1.75E+6	5.97E+2	4.40E-120
	e	1.83E+6	6.37E+2	2.87E+3		
	T	2.29E+7	6.49E+2			

	Sr	SS	df	MS	$\tilde{\chi}^2$	prob > $\tilde{\chi}^2$
F20	*Cl*	2.06E+7	1.20E+1	1.71E+6	5.84E+2	3.49E-117
	e	2.31E+6	6.37E+2	3.62E+3		
	T	2.29E+7	6.49E+2			
F21	*Cl*	1.99E+7	1.20E+1	1.66E+6	5.64E+2	5.65E-113
	e	3.00E+6	6.37E+2	4.72E+3		
	T	2.29E+7	6.49E+2			
F34	*Cl*	1.09E+7	1.20E+1	9.11E+5	3.10E+2	3.81E-59
	e	1.20E+7	6.37E+2	1.88E+4		
	T	2.29E+7	6.49E+2			
F35	*Cl*	2.08E+7	1.20E+1	1.73E+6	5.89E+2	1.94E-118
	e	2.10E+6	6.37E+2	3.30E+3		
	T	2.29E+7	6.49E+2			
F36	*Cl*	1.39E+7	1.20E+1	1.16E+6	3.94E+2	5.74E-77
	e	8.98E+6	6.37E+2	1.41E+4		
	T	2.29E+7	6.49E+2			
F37	*Cl*	2.04E+7	1.20E+1	1.70E+6	5.79E+2	3.63E-116
	e	2.48E+6	6.37E+2	3.89E+3		
	T	2.29E+7	6.49E+2			
F38	*Cl*	2.19E+7	1.20E+1	1.82E+6	6.20E+2	6.09E-125
	e	1.03E+6	6.37E+2	1.61E+3		
	T	2.29E+7	6.49E+2			
F39	*Cl*	1.41E+7	1.20E+1	1.18E+6	4.01E+2	2.28E-78
	e	8.74E+6	6.37E+2	1.37E+4		
	T	2.29E+7	6.49E+2			
F40	*Cl*	1.28E+7	1.20E+1	1.07E+6	3.63E+2	2.50E-70
	e	1.01E+7	6.37E+2	1.58E+4		
	T	2.29E+7	6.49E+2			
F41	*Cl*	1.31E+7	1.20E+1	1.10E+6	3.73E+2	2.12E-72
	e	9.74E+6	6.37E+2	1.53E+4		
	T	2.29E+7	6.49E+2			
F42	*Cl*	6.86E+6	1.20E+1	5.71E+5	1.94E+2	4.55E-35
	e	1.60E+7	6.37E+2	2.52E+4		
	T	2.29E+7	6.49E+2			
F43	*Cl*	9.53E+6	1.20E+1	7.95E+5	2.70E+2	7.52E-51
	e	1.34E+7	6.37E+2	2.10E+4		
	T	2.29E+7	6.49E+2			
F22	*Cl*	1.96E+7	1.20E+1	1.63E+6	5.56E+2	2.86E-111
	e	3.29E+6	6.37E+2	5.16E+3		
	T	2.29E+7	6.49E+2			
F23	*Cl*	2.12E+7	1.20E+1	1.77E+6	6.02E+2	3.68E-121
	e	1.65E+6	6.37E+2	2.59E+3		
	T	2.29E+7	6.49E+2			
F24	*Cl*	1.89E+7	1.20E+1	1.58E+6	5.36E+2	4.45E-107
	e	3.98E+6	6.37E+2	6.25E+3		
	T	2.29E+7	6.49E+2			
F25	*Cl*	2.03E+7	1.20E+1	1.69E+6	5.75E+2	1.83E-115
	e	2.59E+6	6.37E+2	4.07E+3		
	T	2.29E+7	6.49E+2			
F26	*Cl*	1.94E+7	1.20E+1	1.62E+6	5.51E+2	3.10E-110
	e	3.46E+6	6.37E+2	5.43E+3		
	T	2.29E+7	6.49E+2			
F27	*Cl*	2.04E+7	1.20E+1	1.70E+6	5.79E+2	2.93E-116
	e	2.46E+6	6.37E+2	3.86E+3		
	T	2.29E+7	6.49E+2			

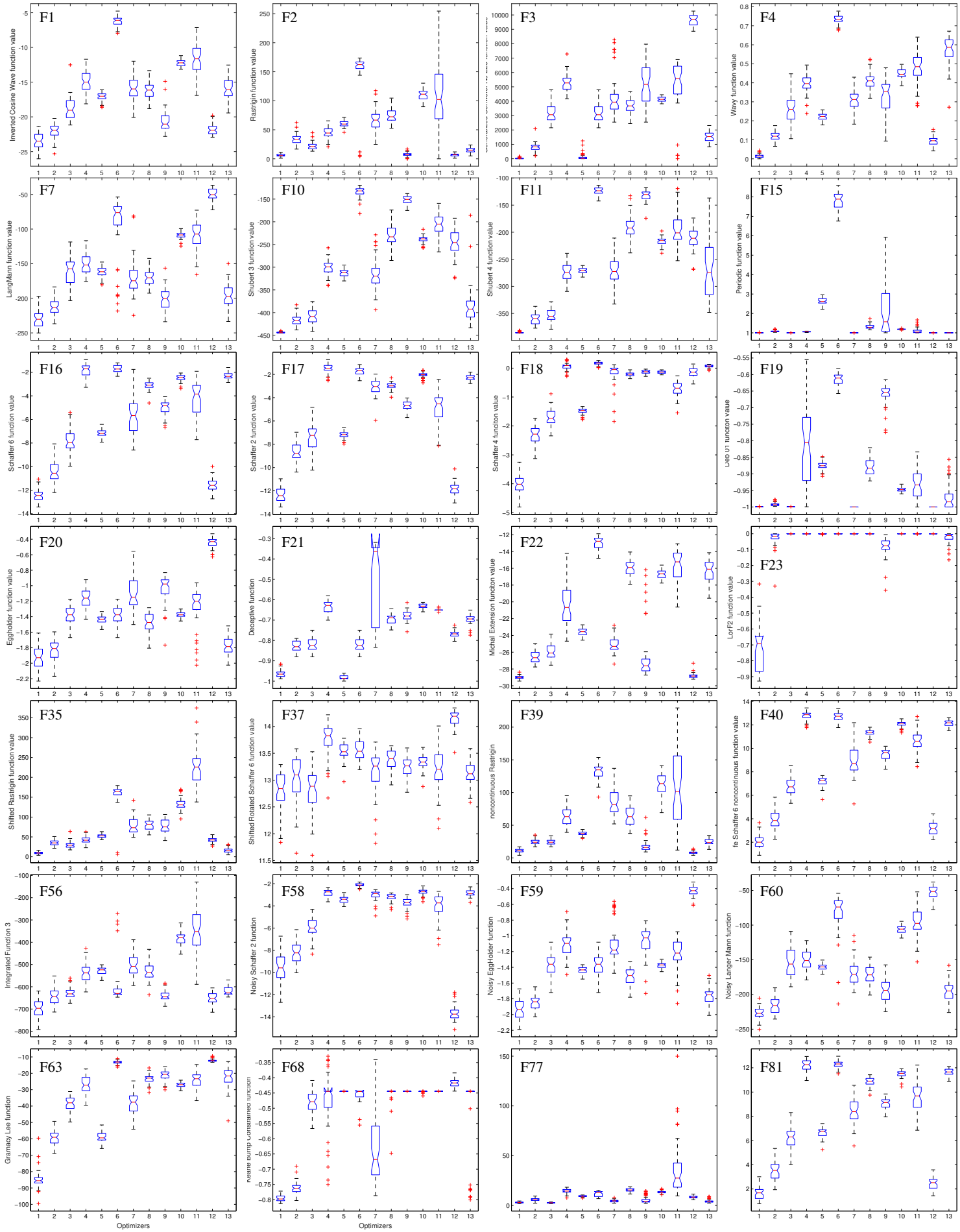


Fig. 12: Statistical Analysis of the Results based on KWT for 30 variables test functions.

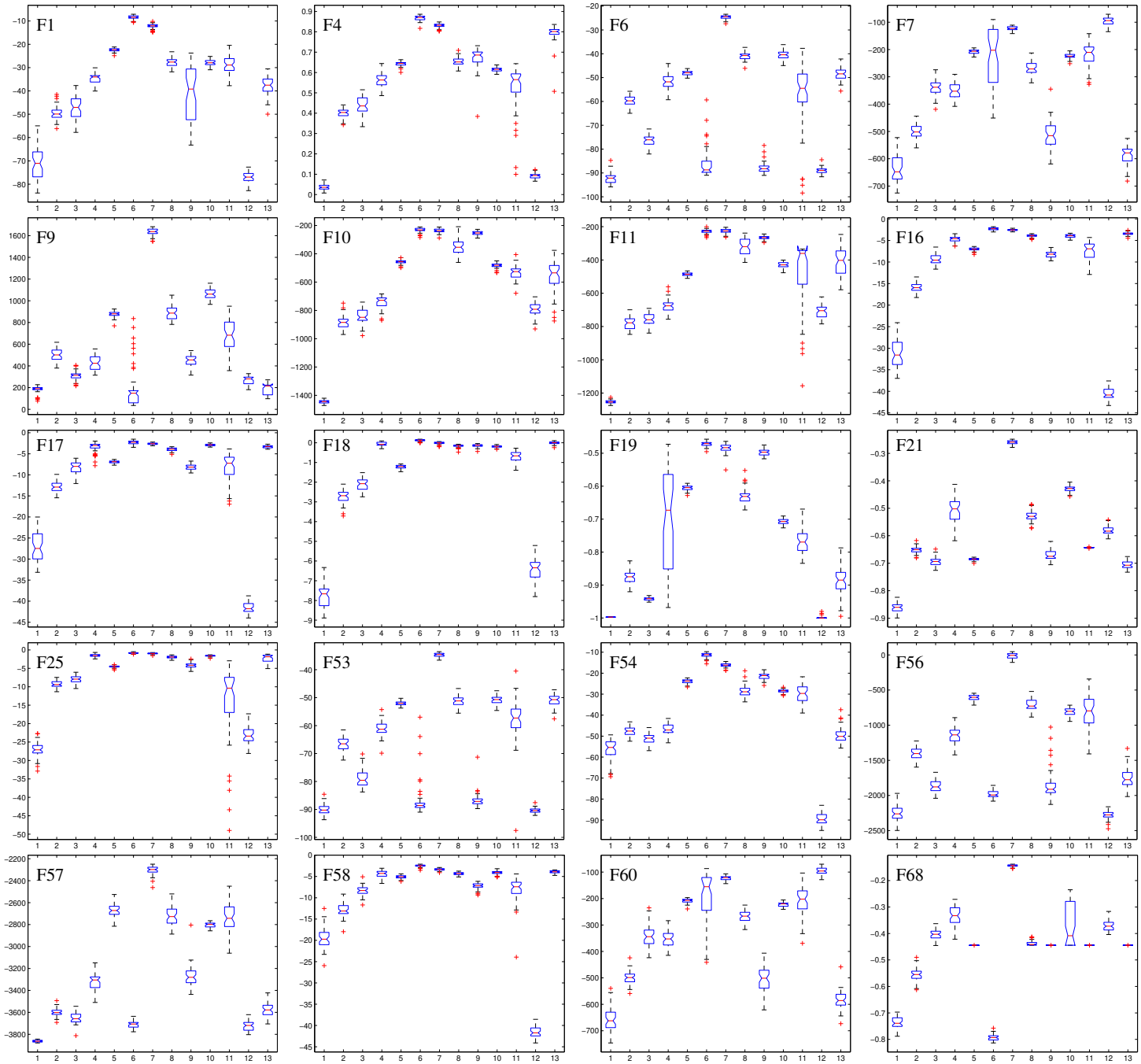


Fig. 13: Statistical Analysis of the Results based on KWT for 100 variables test functions.

noise-free solution. On this function, CMAES provides unfeasible out of bound solution.

F19 On this function, METO appears as the best optimizer. Equal ψ and μ show symmetric distribution of the solution with low uncertainty and high consistency below -29.0. Its worst performance is also better than others.

F20 For this function solution distribution is also symmetric due to equal μ and ψ with low uncertainty $\sigma = 0.2$. The best value of the function is achieved by METO as well, which is -0.93. As usual, consistency below -0.72 is the highest as can be seen in Table-6.

F21 Deformed Schaffer2 is the deformed version of F14 with higher complexity. METO performs best in all performance measures with uncertainty level of $\sigma = 0.8$. Smaller $\psi = -13.8$ than $\mu = -12.5$ shows a left-skewed distribution which is good to achieve high accuracy below -12.5 as shown in Table-6.

F22 We also tested on Keane Bump function, which is a constrained function. No other algorithm can compete with METO for this function except BHGA to some extent. We can observe that METO is very consistent with zero noise $\sigma = 0.1$ on this function. This is one of the very hard functions to solve, where METO shows its importance to adopt as better Evolutionary Algorithm. Comparative results on this function are given in Table-6.

Table 10: Kruskal Wallis ANOVA Table

	Sr	SS	df	MS	$\tilde{\chi}^2$	*prob¿* $\tilde{\chi}^2$
F28	*Cl*	1.35E+7	1.20E+1	1.12E+6	3.82E+2	2.94E-74
	e	9.43E+6	6.37E+2	1.48E+4		
	T	2.29E+7	6.49E+2			
F12	*Cl*	2.06E+7	1.20E+1	1.72E+6	5.85E+2	1.33E-117
	e	2.24E+6	6.37E+2	3.52E+3		
	T	2.29E+7	6.49E+2			
F29	*Cl*	9.08E+6	1.20E+1	7.57E+5	2.58E+2	3.56E-48
	e	1.38E+7	6.37E+2	2.17E+4		
	T	2.29E+7	6.49E+2			
F30	*Cl*	1.61E+7	1.20E+1	1.35E+6	4.58E+2	1.90E-90
	e	6.74E+6	6.37E+2	1.06E+4		
	T	2.29E+7	6.49E+2			
F31	*Cl*	1.99E+7	1.20E+1	1.66E+6	5.66E+2	2.27E-113
	e	2.94E+6	6.37E+2	4.61E+3		
	T	2.29E+7	6.49E+2			
F32	*Cl*	2.19E+7	1.20E+1	1.83E+6	6.22E+2	2.66E-125
	e	9.68E+5	6.37E+2	1.52E+3		
	T	2.29E+7	6.49E+2			
F33	*Cl*	1.34E+6	1.20E+1	1.12E+5	3.81E+1	0.000146335
	e	2.15E+7	6.37E+2	3.38E+4		
	T	2.29E+7	6.49E+2			

F23 This function is complicated to solve due to its multi-modality. Best solution for this function is achieved by METO as $\mathbb{B} = -29.0$ with best $\mu = -28.0$as shown in Table-7.

F24 For this function, again METO gives better average performance with high consistency. The best solution is also achieved by METO.

F25 On this function, METO performs the best whis no other optimizer can compete it. Due to its left-skewed distribution achieved consistency is very high as shown in Table-7.

F26 We also tested the Noisy functions, one of them is Schaffer2, on this function METO achieved the second position after GSA.

F27 On the second Noisy function, METO outperforms in all statistical parameters. Here, with best average performance, it is achieving highest consistency below the threshold value of -19297 due to left-skewed distribution where $(\psi = -19371) < (\mu = -19297)$.

F28 On LangerMann Noisy function METO yet performs better than other optimizers with very high consistency. The best solution is also achieved by METO as shown in Table-7.

F29 Noisy version of Hybrid Composition Functions of F39 is also tested. Where, METO secures 4^{th} position after CMAES, GSA and SFLA. No other algorithm can compete with CMAES algorithm for this function.

F30 On the noisy version of Rotated Hybrid Composition Function F42, METO secures 3rd position again after SLPSO and BA algorithms.

F31 On Expanded Extended Griewank's plus Rosenbrock function, METO is backed by CMAES and BBO algorithms, where the best value of this function achieved by METO and BBO are same as 1.8.

F32 On this function, METO is best as usual and with low uncertainty, $\sigma = 0.6$. On this function best solution is also achieved by METO which is $\mathbb{B} = 0.8$. No other algorithm are can find even a near solution, even average performance of METO $\mu = 1.7$ is better than the best performance of the other optimizers on this function.

F33 All optimizers gives a similar performance on this function.

F34 On Rotated Ackley function performance of the METO is after GSA. However, METO, GA, BBO, and BA are giving similar performance.

F35 METO provides a better solution for the Rotated Rastrigin function with best average performance of $\mu = 9.5$ with high consistency. Also, uncertainty $\sigma = 2.6$ is minimum.

F36 On Rotated ScafferF6 function, BBO and METO are giving similar results.

F37 METO performance on the non-continuous functions is outstanding. On this function, METO has good performance backed by GSA where the achieved best solution by both optimizers are same.

F38 On fE-ScafferF6 non-continuous function, no other optimizer can compete the METO. Left-skewed distribution of solution gives the best consistency level.

F39 On this function, which is the composition of 5 functions, DE perform better than METO.

F40 GSA performs the best on this function, which is rotated version of the previous function. METO secures the second position for this function with lower uncertainty level $\sigma = 15.8$ than other optimizers.

F41 On Noisy version of the previous function, METO persists the second position after GSA with comparatively low $\sigma = 22.5$. The worst performance of the METO is best among all optimizers.

F42 Hybrid Rotated Composition Function 3 is solved best by SLPSO optimizer. Performance of METO is average on this function.

F43 Non-Continuous Hybrid Rotated Composition Function 3 is solved best by CMAES algorithm, where the performance of METO is average on this function.

Overall, the results shown in Tables 5 to 8 prove the dominance of METO over other optimizers for multi-modal problems with single or few global extremes.

5.2 Kruskal Wallis statistical analysis of the results

We tested distributions of all optimizers for their significant difference using KWT one way ANOVA rank test. It is an extended version of the Mann-Whitney test. For the test, we consider the Null hypothesis, H_0, as "distribution of METO is same as the distribution of other optimizers", where distributions of all optimizers are coming from independent experiments. It discriminates the distribution of all optimizer based on the calculated "critical chi-square value $\tilde{\chi}^2$" and KWT value. If the value of KWT is smaller than the $\tilde{\chi}^2$, the H_0 cannot be rejected. Thus, to reject the H_0, KWT value should be greater than $\tilde{\chi}^2$. For this procedure, p-value is utilized to test the significant difference between distributions with 1% significance level.

ANOVA Tables- 9 and 10 provide additional test results. ANOVA results for each function has six attributes. Sr represents the source of the variability. Based on the different type of variability here, three types of sources are given. First is *Cl*, representing groups, it is due to the variability that exists due to the differences among the distribution means. Second is *e*, which is error, and the variability exists due to the differences between the data set within the group and the group mean. This is also called variability within the group. The third is the *T* Total, which represents total variability. SS is the sum of square due to each Sr, df is the degree of freedom, df associated with each Sr is calculated. For *Cl*, df is the degree of freedom (DoF) between the distributions/groups and calculated as df =$K-1$; here $K=13$ the number of the optimizers. For *e*, the df is the DoF within the distribution groups and defined as df = $N-K$, here $N=650$, the number of observations. The total DoF is calculated as df = $N-1$, which is equal to $(N-K)+(K-1)$. Next attribute, MS is the mean squares for each source and calculated as $\frac{\text{SS}}{\text{df}}$. F-statistics, which is ζ, represented for the Sr and the ratio of the MS. The last column of this table is p-value, which is the probability that the $\tilde{\chi}^2$can take a value larger than the computed test-statistic value. ANOVA1 derives this probability from the cdf of F-distribution [57]. In the ANOVA table, $\tilde{\chi}^2$ and p-value are important, where other above-described parameters value support to calculate them.

Moreover, to show the significant difference between the distributions of solutions achieved by each optimizer, notched box-plot is shown in Fig 12. The notched box is associated with an optimizer which has two sections divided by a centerline, and this is the median. Two end edges of each notched box, the bottom and the top, indicates the $q_i = 25^{th}$ and $q_3 = 75^{th}$ percentiles, respectively. Outliers O in the distribution are plotted individually using the '+' symbol. In Fig. 12, we can observe that METO results are free of any outlier. Also, therer is a significant difference between the METO and other optimizers. We can observe that notches of METO box plot do not overlap the others, which shows that true medians do differ with others with 95% confidence level. Beyond the whiskers length, the i^{th} solution in the solution distribution are displayed as outliers O_i: if $O_i > q_3 + w(q_3-q_1)$ or $O_i < q_1-w(q_3-q_1)$, where w is the maximum whisker length. Horizontal axes numbers in each sub-plot represent optimizer number, where from 1 to 13 it are respectively METO, BHGA, BBO, IWO, DE, CMAES, SFLA, FA, TLBO, CUCKOO, BA, GSA, SLPSO, respectively.

5.3 Statistical ranking

In this manuscript, significant results are presented due to the limitation of pages. It is worth to discuss the features and limitation of the METO in this section based on the experimental results and associated Kruskal Wallis statistical test. The rank-sum scores of optimizers for all functions are given in Table 11. Based on the score in the table, we can point out that for how many functions METO secures based rank score. Accordingly, we can rank all the optimizer from the best rank to the worst rank. In Table 12, we can see that on 24 functions METO secured the first position (P_1), on 7 functions it is on the second position (P_2), as so on. The rank count is based on the Table-11.

For getting a numerical value, we adopted the scoring algorithm based on the average performance. We assign 13 points to the best optimizer and reduces it by 1 for each degraded performer, sequentially. This is because we have $K=13$ optimizers in this manuscript. This scoring is averaged after applying for all benchmark functions. Optimizer with the highest score is best ranked. In this paper, we show the results only for the functions presented in Tables 3 AND 4. The $\omega = \sum_{i=1}^{K}(P_i \times (14-i)))$ is the aggregated score to show overall performance of all optimizers, for example, METO has 502 score which is at 1^{st} position, similarly, BHGA is at 2^{nd} position with 419 scores as so on. Finaly, average score is calculated by Average $-\ \omega = \omega/K$. The optimizer with the highest Average is considered as winner.

For the Table-12, we can see on-average METO is flagged as one of the best optimizers by achieved high average score. Ranking of optimizers for 30 variable problems can be observed from high to low score as: METO, BHGA, BBO, TLBO, FA, SLPSO, CMAES, CUCKOO, SFLA, GSA, BA, DE and IWO in the sequence.

5.4 Observation and discussion on 100 variables problems

We motivated to test the performance of the METO for more variables problems to see its persistent goodness. We solved the 100 variables benchmark problems and statistically tested as well using the Kruskal Wallis test. Tables 13, and 14 show the results on selected benchmark functions from the Table-3 and 4 on which METO performance is significantly

Table 11: Kruskal Wallis Rank sum score

	METO	BHGA	BBO	IWO	DE	CMAES	SFLA	FA	TLBO	CUCKOO	BA	GSA	SLPSO
F1	41	98	238	428	311	625	382	368	168	541	543	113	376
F2	70	290	230	340	410	586	430	469	99	553	492	79	183
F3	40	129	289	512	66	289	392	340	479	405	489	626	177
F4	26	118	248	390	206	626	290	407	314	471	489	89	558
F5	155	283	169	425	337	458	442	484	137	623	526	123	70
F6	38	94	336	388	338	517	280	290	145	505	506	624	170
F7	68	220	218	211	473	576	394	398	245	521	587	202	119
F8	26	103	123	300	267	621	264	448	579	429	498	411	162
F9	26	94	109	246	259	610	256	482	586	402	467	422	273
F10	173	329	217	402	586	63	271	353	301	537	583	275	141
F11	223	357	191	328	560	626	40	487	464	439	329	125	62
F12	26	98	203	355	102	582	213	416	457	330	395	617	436
F13	32	117	195	578	225	600	294	415	324	489	369	78	514
F14	39	137	193	601	203	575	402	401	301	529	302	62	486
F15	26	80	135	534	165	609	415	324	389	386	228	401	541
F16	160	257	180	477	463	619	28	453	573	350	347	84	242
F17	53	86	322	476	252	322	490	241	523	325	414	625	103
F18	66	171	184	539	35	184	517	382	429	562	504	280	380
F19	40	181	213	391	316	623	256	506	171	465	518	63	488
F20	26	146	297	499	219	612	453	337	91	454	485	466	148
F21	100	186	246	570	271	613	135	461	425	538	72	295	321
F22	100	270	226	360	376	483	452	457	156	611	549	102	89
F23	86	132	216	304	421	626	254	415	544	498	493	31	212
F24	57	173	217	423	446	264	475	436	168	590	596	137	252
F25	88	116	179	492	324	624	439	377	281	513	294	26	479
F26	50	77	324	493	262	324	485	217	511	328	426	625	108
F27	39	79	346	379	332	547	278	271	165	495	523	620	158
F28	192	338	219	185	494	433	185	339	311	493	610	99	335
F29	331	315	421	245	353	31	416	325	313	325	283	585	288
F30	30	83	250	421	475	239	202	489	399	370	364	610	301
F31	84	247	54	518	365	432	183	553	214	480	602	333	165
F32	33	118	193	586	213	597	298	425	327	493	357	76	516
F33	325	324	318	358	338	313	384	334	340	304	186	365	343
F34	220	327	186	241	419	335	476	455	446	413	237	30	448
F35	34	190	153	246	318	544	421	433	420	531	622	244	76
F36	114	215	135	526	454	456	262	366	273	331	273	619	209
F37	75	228	224	424	330	597	485	424	143	551	494	37	219
F38	31	112	192	597	213	586	300	412	316	497	381	84	509
F39	159	313	333	312	43	342	404	142	379	373	619	529	284
F40	202	327	233	189	494	442	240	339	234	480	605	95	351
F41	213	323	204	187	485	417	202	341	346	483	603	73	353
F42	327	306	230	265	495	272	227	309	242	503	342	501	213
F43	321	279	454	255	336	26	404	334	323	332	319	584	267

Table 12: Rank of all optimizer based on Kruskal Wallis Rank Sum scores: 30 variables

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}	Sum	ω	Average-ω	Winner
METO	**23**	**7**	**5**	**2**	**2**	**2**	0	0	**2**	0	0	0	0	43	**502**	**38.6**	**1st**
BHGA	0	10	10	6	1	11	2	3	0	0	0	0	0	43	419	32.2	2^{nd}
BBO	1	2	5	13	14	3	2	1	0	0	0	2	0	43	396	30.5	3^{rd}
IWO	0	5	0	2	3	2	5	5	2	4	6	7	2	43	248	19.1	13^{th}
DE	2	1	1	2	7	4	6	5	2	6	2	4	1	43	285	21.9	12^{th}
CMAES	3	0	1	1	1	2	4	1	1	4	2	7	16	43	176	13.5	7^{th}
SFLA	2	1	4	1	6	8	0	3	7	3	6	0	2	43	295	22.7	9^{th}
FA	0	1	0	2	1	2	5	7	13	5	4	3	0	43	237	18.2	5^{th}
TLBO	0	1	6	4	4	2	8	3	4	4	1	6	0	43	295	22.7	4^{th}
CUCKOO	0	1	0	0	0	1	2	8	6	7	10	4	4	43	182	14.0	8^{th}
BA	2	0	1	1	2	2	4	4	2	4	8	5	8	43	201	15.5	11^{th}
GSA	7	10	4	0	1	2	2	1	2	1	0	3	10	43	330	25.4	10^{th}
SLPSO	3	4	6	9	1	2	3	2	2	5	4	2	0	43	347	26.7	6^{th}
Sum	43	43	43	43	43	43	43	43	43	43	43	43	43				
MF	13	12	11	10	9	8	7	6	5	4	3	2	1				

outstanding. All algorithms run a hundred times with the same parameters as above for all optimizers. Based on the output distribution of all algorithms, a comparison table is formed based on their mean μ, standard deviation σ, best value achieved $\mathbb{B}$, worst performance $\mathscr{W}$ and consistency $\mathscr{C}$. Consistency in percentage shows how many time a specific algorithm achieves solution better than a threshold, which is the mean value of METO. From the results, we can observe that METO is best in all measures for functions F1, F3, F4, F6, F8, F9, F15, F44, and F45. On these functions, no other algorithm competes with METO. Moreover, for F7 METO is giving the best average performance but CMAES achieves the best performance and higher consistency. GSA is best for F13, F14, F24, and F26. On F18 and F22 CMAES is best. For other functions, results are mix where METO appears as a second-best performer.

Based on the output distribution by several independent runs of all algorithms, the results of Kruskal Wallis test are presented in Table-15. In this table, chi-square value $\tilde{\chi}^2$ and calculated p-value is presented. The smaller p-value than $\tilde{\chi}^2$ shows that the solution distribution of METO is significantly different from other optimizers. To see the performance of METO on multiple objective problem we includes 100 variables Kursawe funciton as mentioned F44[6], with linear sum of all objectives. On this problem, we can see that METO is giving best performance in all matrices, highlighted in Table-14. Moreover, METO performance is significantly observable on two more functions, noisy langerman, F45= noisy(F6) and Xinshe Yangn function number-4, F46[7]. Table-16 shows the Kruskal Wallis rank sum score for the selected functions. Table-17 is the output of this table. Rows of Table-17 represents the algorithms, where each column is position P_K, where $K = 1,2,\ldots,13$. We can observe that METO is on P_1 position 11 times and on P_2 position 11 times. It shows that METO performance is good comparative to other optimizers. Overall, METO has $\omega = 275$ the average of it is Average-$\omega = 21.2$, which is maximum of all. Thus METO is placed at 1^{st} position, similarly, BBO is at 2^{nd} position with $\omega = 208$ and so on. For the Table 17, we can observe that METO is flagged as one of the best optimizers with 11 best and 11 good performances. Ranking of the optimizers from high to low average-ω score is as: METO, BBO, IWO, FA, CUCKOO, TLBO, SLPSO, GSA, SFLA, BA, CMAES, BHGA, and DE in the sequence. We can observe that for higher variables problems BHGA degrades its performance.

It is observed from Table 13 and 14 that METO may be an alternative optimizer to solve the undertaken functions with good average performance. On Functions F3, F6, F9, F10 and F28, METO is the best in all aspects, where it achieves higher accuracy, best value , minimum worst value with low uncertainty in the results compared to other algorithms. METO shows reasonable and comparable consistency.

We have also tested with many variants of GA as given in Ref. [40], and deduce the observation based on the simulation results that, METO is a good replacement of GA for highly complex problems. It works best on single global solution problems with multiple local solutions.

[6] F44= $f_1 + f_2$ in the range $x_i = [-5,5]$ $f_1 = \sum_{i=1}^{n}\left[-10\exp\left(-0.2\sqrt{x_i^2+x_{i+1}^2}\right)\right]$ and $f_2 = \sum_{i=1}^{n}\left[|x_i|^{0.8}+5\sin\left(x_i^3\right)\right]$

[7] F46 $= \left[\sum_{i=1}^{n}{}^2(x_i) - e^{-\sum_{i=1}^{n}x_i^2}\right] e^{-\sum_{i=1}^{n}\sin^2\sqrt{|x_i|}}$, where, $x_i = [-10,10]$

Table 13: Statistical results on 100 variables problems

		METO	BHGA	BBO	IWO	DE	CMAES	SFLA	FA	TLBO	CUCKOO	BA	GSA	SLPSO
F1	μ	**-77.08**	-49.50	-47.05	-34.59	-22.37	-8.50	-12.16	-27.84	-41.70	-27.88	-28.72	-77.01	-37.79
	σ	**6.87**	2.96	4.94	2.46	0.73	0.86	0.96	2.08	12.76	1.23	3.92	1.95	4.02
	$\mathbb{B}$	**-83.80**	-56.14	-57.76	-40.05	-24.88	-10.68	-14.89	-31.80	-63.22	-30.92	-37.76	-82.78	-49.96
	$\mathscr{W}$	**-54.98**	-41.47	-37.71	-30.08	-21.15	-7.10	-10.18	-23.22	-23.77	-25.22	-20.45	-72.66	-30.54
	$\mathscr{C}$	**86%**	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	80%	0%
F3	μ	**1720.4**	11455.6	17387.8	22048.8	23704.7	17387.8	33096.4	23269.8	25113.4	21655.3	19813.0	36404.8	6286.4
	σ	**386.50**	1027.09	1117.72	1264.52	460.43	1117.72	696.34	1079.96	5134.97	352.44	6564.18	789.35	873.22
	$\mathbb{B}$	**606.89**	8480.95	14658.78	19872.87	22479.04	14658.78	30217.77	20855.21	17685.32	20846.91	1527.86	33467.22	4503.69
	$\mathscr{W}$	**2537.6**	13778.4	19739.5	24381.6	24524.5	19739.5	34141.3	25982.5	33071.0	22420.5	23839.4	37639.4	8239.1
	$\mathscr{C}$	**50%**	0%	0%	0%	0%	0%	0%	0%	0%	0%	2%	0%	0%
F4	μ	**0.0361**	0.3992	0.4403	0.5676	0.6428	0.8676	0.8302	0.6533	0.6728	0.6152	0.5233	0.0923	0.7917
	σ	**0.0140**	0.0225	0.0453	0.0405	0.0113	0.0127	0.0105	0.0194	0.0535	0.0096	0.1174	0.0135	0.0469
	$\mathbb{B}$	**0.0074**	0.3426	0.3337	0.4863	0.6004	0.8171	0.8034	0.6073	0.3841	0.5912	0.0996	0.0657	0.5073
	$\mathscr{W}$	**0.0717**	0.4406	0.5150	0.6440	0.6629	0.8866	0.8483	0.7086	0.7317	0.6373	0.6436	0.1231	0.8361
	$\mathscr{C}$	**50%**	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
F5	μ	-92.24	-59.70	-76.31	-51.71	-48.09	-86.11	-24.62	-40.90	-87.95	-40.57	-57.27	-88.95	-48.54
	σ	2.17	2.02	2.37	2.96	0.87	6.34	0.85	1.53	2.34	1.94	13.71	1.21	2.62
	$\mathbb{B}$	-95.86	-64.96	-82.06	-59.29	-50.24	-90.99	-27.40	-46.10	-91.04	-44.95	-98.44	-91.63	-55.65
	$\mathscr{W}$	-84.75	-55.75	-71.55	-44.09	-46.22	-59.38	-23.40	-37.33	-78.46	-36.18	-37.72	-84.48	-42.20
	$\mathscr{C}$	50%	0%	0%	0%	0%	0%	0%	0%	0%	0%	8%	0%	0%
F6	μ	**-638.82**	-497.98	-339.99	-352.75	-208.05	-226.29	-122.37	-267.95	-512.78	-223.89	-217.70	-95.52	-585.58
	σ	**47.84**	23.86	29.78	29.27	8.11	105.95	7.44	22.63	56.46	8.65	42.73	15.25	39.09
	$\mathbb{B}$	**-725.63**	-560.16	-418.89	-408.32	-227.56	-451.05	-141.70	-321.89	-619.18	-251.68	-327.88	-134.55	-681.76
	$\mathscr{W}$	**-522.55**	-444.29	-274.33	-290.76	-193.94	-90.23	-110.69	-212.78	-345.07	-205.02	-142.00	-70.70	-525.62
	$\mathscr{C}$	**58%**	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	10%
F7	μ	183.9	501.0	311.0	427.3	876.7	200.2	1635.1	889.7	449.7	1062.7	681.0	265.8	191.2
	σ	33.1	57.4	40.3	63.9	25.8	192.4	31.0	67.5	53.1	44.7	142.9	41.4	50.7
	$\mathbb{B}$	76.4	380.2	215.7	314.9	769.6	34.9	1547.2	782.7	315.0	967.1	356.7	181.1	97.5
	$\mathscr{W}$	226.8	617.8	406.0	556.5	924.3	836.6	1682.9	1051.7	541.4	1161.9	950.3	328.8	272.0
	$\mathscr{C}$	32%	0%	0%	0%	0%	76%	0%	0%	0%	0%	0%	4%	30%
F8	μ	**-1444.6**	-884.8	-842.7	-745.3	-456.6	-232.2	-236.9	-345.1	-254.4	-482.8	-533.6	-795.7	-553.0
	σ	**11.4**	46.6	49.8	47.0	15.3	15.1	15.1	60.6	13.9	16.7	48.6	48.7	111.6
	$\mathbb{B}$	**-1472.2**	-970.2	-978.3	-870.5	-499.1	-284.8	-288.0	-460.8	-288.8	-534.5	-678.8	-930.8	-874.5
	$\mathscr{W}$	**-1420.7**	-748.7	-741.7	-683.6	-427.6	-210.4	-210.9	-208.5	-227.1	-450.2	-405.7	-703.9	-375.7
	$\mathscr{C}$	**50%**	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
F9	μ	**-1253.2**	-779.6	-758.3	-676.1	-486.2	-228.1	-226.9	-319.8	-265.5	-429.3	-465.7	-709.6	-409.1
	σ	**9.1**	36.2	39.5	34.7	11.0	12.0	13.5	50.4	10.8	16.9	196.4	40.4	87.9
	$\mathbb{B}$	**-1275.0**	-847.1	-839.6	-756.2	-510.6	-265.3	-263.7	-414.6	-295.9	-476.8	-1156.3	-784.3	-578.9
	$\mathscr{W}$	**-1225.8**	-699.4	-691.2	-561.5	-465.8	-200.0	-203.5	-238.5	-243.9	-401.0	-334.3	-622.0	-246.5
	$\mathscr{C}$	**52%**	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
F13	μ	-31.06	-15.92	-9.40	-4.74	-7.03	-2.30	-2.56	-3.90	-8.26	-3.96	-7.54	-40.51	-3.42
	σ	3.81	1.17	1.10	0.70	0.36	0.30	0.17	0.30	0.71	0.39	2.14	1.27	0.37
	$\mathbb{B}$	-37.00	-18.23	-11.66	-6.32	-8.25	-3.03	-3.01	-4.81	-9.72	-4.94	-12.89	-43.29	-4.48
	$\mathscr{W}$	-24.08	-13.47	-6.53	-3.44	-6.40	-1.81	-2.17	-3.42	-6.63	-3.33	-4.31	-37.59	-2.64
	$\mathscr{C}$	58%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%	0%
F14	μ	-26.91	-12.72	-8.33	-3.47	-6.92	-2.31	-2.62	-3.97	-8.17	-2.91	-8.18	-41.50	-3.32
	σ	3.55	1.27	1.30	1.24	0.30	0.44	0.23	0.40	0.71	0.24	3.20	1.29	0.31
	$\mathbb{B}$	-33.15	-15.45	-12.06	-7.83	-7.71	-3.45	-3.10	-5.14	-9.56	-3.42	-16.95	-43.99	-3.92
	$\mathscr{W}$	-20.02	-9.89	-6.06	-1.99	-6.34	-1.53	-2.21	-3.32	-6.74	-2.41	-3.86	-38.73	-2.72
	$\mathscr{C}$	52%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%	0%
F15	μ	**-7.720**	-2.743	-2.130	-0.063	-1.219	0.120	-0.013	-0.165	-0.141	-0.192	-0.717	-6.418	-0.012
	σ	**0.627**	0.349	0.296	0.103	0.091	0.044	0.057	0.069	0.076	0.052	0.297	0.597	0.068
	$\mathbb{B}$	**-8.890**	-3.715	-2.740	-0.301	-1.468	-0.006	-0.199	-0.474	-0.441	-0.331	-1.392	-7.804	-0.247
	$\mathscr{W}$	**-6.331**	-2.100	-1.516	0.093	-1.074	0.183	0.062	-0.075	-0.039	-0.088	-0.276	-5.211	0.108
	$\mathscr{C}$	**46%**	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	4%	0%

Table 14: Statistical results on 100 variables problems

		METO	BHGA	BBO	IWO	DE	CMAES	SFLA	FA	TLBO	CUCKOO	BA	GSA	SLPSO
F16	μ	-0.997	-0.876	-0.942	-0.708	-0.605	-0.472	-0.485	-0.631	-0.497	-0.707	-0.769	-0.998	-0.886
	σ	0.000	0.018	0.004	0.156	0.008	0.007	0.014	0.023	0.011	0.009	0.036	0.005	0.044
	$\mathbb{B}$	-0.998	-0.920	-0.952	-0.968	-0.628	-0.495	-0.551	-0.672	-0.517	-0.727	-0.834	-1.000	-0.995
	$\mathscr{W}$	-0.996	-0.827	-0.932	-0.473	-0.591	-0.457	-0.464	-0.552	-0.475	-0.690	-0.670	-0.980	-0.787
	$\mathscr{C}$	58%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	82%	0%
F18	μ	-0.863	-0.652	-0.693	-0.505	-0.685	#NAME?	-0.260	-0.528	-0.670	-0.429	-0.643	-0.579	-0.707
	σ	0.016	0.012	0.017	0.051	0.004	NaN	0.007	0.020	0.021	0.010	0.002	0.015	0.014
	$\mathbb{B}$	-0.899	-0.681	-0.726	-0.618	-0.701	#NAME?	-0.278	-0.572	-0.705	-0.457	-0.646	-0.611	-0.733
	$\mathscr{W}$	-0.824	-0.617	-0.649	-0.413	-0.678	#NAME?	-0.248	-0.486	-0.621	-0.404	-0.640	-0.542	-0.676
	$\mathscr{C}$	44%	0%	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%	0%
F21	μ	-27.11	-9.31	-7.96	-1.54	-4.53	-0.84	-0.97	-1.98	-4.15	-1.62	-15.16	-23.22	-2.36
	σ	2.06	0.99	0.99	0.38	0.27	0.16	0.16	0.36	0.69	0.17	12.41	2.25	1.17
	$\mathbb{B}$	-32.88	-11.34	-10.47	-2.42	-5.43	-1.18	-1.54	-2.82	-5.82	-2.29	-49.00	-28.09	-5.06
	$\mathscr{W}$	-22.70	-7.46	-6.05	-0.68	-3.97	-0.47	-0.72	-1.34	-2.45	-1.33	-2.96	-17.34	-1.16
	$\mathscr{C}$	0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.14	0.06	0.00
F23	μ	-89.97	-66.60	-78.97	-61.21	-52.02	-86.72	-34.74	-51.38	-86.76	-50.80	-57.73	-90.43	-50.87
	σ	1.97	2.41	3.16	2.65	0.81	6.43	0.84	1.88	2.62	1.43	8.20	0.87	2.13
	$\mathbb{B}$	-93.69	-72.34	-83.74	-69.87	-53.72	-90.96	-36.59	-55.55	-89.71	-54.60	-97.53	-92.13	-57.57
	$\mathscr{W}$	-84.56	-61.54	-70.15	-54.26	-50.23	-56.98	-33.54	-46.73	-71.30	-47.53	-40.48	-87.56	-47.19
	$\mathscr{C}$	58%	0%	0%	0%	0%	6%	0%	0%	0%	0%	2%	72%	0%
F24	μ	-56.64	-47.76	-51.22	-46.94	-23.92	-11.37	-16.11	-28.52	-21.45	-28.51	-30.03	-89.53	-49.35
	σ	4.98	1.97	2.54	2.58	1.01	1.07	0.90	2.78	1.50	0.83	3.86	2.54	3.79
	$\mathbb{B}$	-69.35	-52.43	-56.91	-53.17	-26.60	-15.55	-18.76	-33.67	-25.86	-30.68	-38.98	-94.97	-55.69
	$\mathscr{W}$	-49.41	-43.20	-45.92	-41.59	-22.24	-9.74	-14.42	-18.85	-18.35	-26.66	-21.71	-82.98	-37.40
	$\mathscr{C}$	34%	0%	2%	0%	0%	0%	0%	0%	0%	0%	0%	100%	0%
F25	μ	-2254.4	-1405.5	-1860.9	-1149.2	-606.0	-1983.4	-13.0	-712.1	-1854.1	-809.5	-806.2	-2281.0	-1756.9
	σ	108.4	89.5	94.8	110.9	40.0	54.9	38.9	77.0	215.7	53.0	245.5	59.0	147.3
	$\mathbb{B}$	-2498.9	-1597.1	-2040.1	-1421.8	-714.5	-2081.2	-106.3	-884.4	-2127.4	-946.2	-1408.4	-2474.2	-2016.0
	$\mathscr{W}$	-1970.4	-1225.2	-1670.3	-892.7	-544.7	-1856.6	47.1	-519.2	-1028.1	-714.6	-342.5	-2161.1	-1330.5
	$\mathscr{C}$	52%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	68%	0%
F44	μ	**-3862.4**	-3598.9	-3654.5	-3321.9	-2669.5	-3710.9	-2303.7	-2718.7	-3268.8	-2802.3	-2737.5	-3722.7	-3579.5
	σ	**9.9**	34.4	49.6	78.1	58.4	36.2	38.9	81.4	98.8	21.3	132.4	47.3	67.0
	$\mathbb{B}$	**-3879.0**	-3692.0	-3812.7	-3509.8	-2814.7	-3777.1	-2462.0	-2886.7	-3436.5	-2857.1	-3060.4	-3803.1	-3705.3
	$\mathscr{W}$	**-3843.7**	-3493.5	-3544.7	-3148.7	-2525.7	-3635.9	-2248.9	-2520.7	-2804.3	-2765.1	-2450.0	-3621.2	-3422.5
	$\mathscr{C}$	**52%**	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
F26	μ	-19.58	-12.96	-8.29	-4.45	-5.09	-2.48	-3.35	-4.31	-7.30	-4.07	-8.09	-41.56	-3.89
	σ	2.69	1.55	1.16	0.88	0.41	0.27	0.28	0.29	0.74	0.45	3.07	1.18	0.32
	$\mathbb{B}$	-25.93	-17.96	-11.68	-6.62	-6.16	-3.52	-4.09	-5.10	-9.37	-5.14	-23.95	-44.11	-4.74
	$\mathscr{W}$	-12.52	-9.14	-5.07	-3.06	-4.38	-2.09	-2.88	-3.74	-6.13	-3.17	-4.39	-38.55	-3.45
	$\mathscr{C}$	56%	0%	0%	0%	0%	0%	0%	0%	0%	0%	2%	100%	0%
F45	μ	-655.9	-499.0	-343.3	-352.4	-208.7	-193.5	-123.1	-268.4	-507.0	-222.7	-209.7	-95.4	-585.3
	σ	46.2	24.4	41.8	30.3	8.6	98.3	8.0	23.2	48.0	8.5	55.3	13.4	34.1
	$\mathbb{B}$	-746.2	-559.2	-423.7	-414.7	-239.2	-441.2	-143.7	-317.3	-621.0	-240.6	-369.3	-128.6	-673.0
	$\mathscr{W}$	-539.6	-424.2	-235.1	-283.6	-196.4	-86.6	-106.6	-224.8	-406.5	-205.3	-103.6	-69.8	-458.5
	$\mathscr{C}$	56%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	2%
F22	μ	-0.737	-0.556	-0.404	-0.335	-0.444	-0.793	-0.144	-0.438	-0.444	-0.365	-0.444	-0.370	-0.444
	σ	0.021	0.027	0.018	0.039	0.000	0.012	0.004	0.009	0.000	0.084	0.000	0.022	0.000
	$\mathbb{B}$	-0.788	-0.613	-0.445	-0.421	-0.444	-0.814	-0.155	-0.444	-0.444	-0.444	-0.444	-0.404	-0.444
	$\mathscr{W}$	-0.697	-0.490	-0.363	-0.271	-0.444	-0.758	-0.138	-0.413	-0.444	-0.235	-0.444	-0.316	-0.444
	$\mathscr{C}$	54%	0%	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%	0%
F46	μ	9.3E-43	1.3E-39	4.5E-42	1.0E-39	1.1E-31	1.1E-27	1.5E-27	2.6E-32	1.6E-29	1.4E-35	4.2E-39	3.0E-41	6.8E-44
	σ	4.1E-43	1.4E-39	1.1E-42	1.8E-39	7.1E-32	1.8E-27	1.3E-27	5.3E-32	3.8E-29	8.3E-36	1.1E-38	6.3E-41	1.9E-43
	$\mathbb{B}$	2.8E-43	2.1E-40	2.9E-42	2.5E-42	4.8E-33	1.6E-42	7.6E-29	1.6E-34	4.6E-41	2.4E-36	4.3E-42	1.1E-42	3.6E-53
	$\mathscr{W}$	2.4E-42	6.3E-39	8.8E-42	9.6E-39	3.1E-31	9.0E-27	5.8E-27	2.4E-31	2.4E-28	4.3E-35	5.6E-38	3.6E-40	1.2E-42
	$\mathscr{C}$	48%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	98%

Table 15: Kruskal Wallis Test Table

	$\tilde{\chi}^2$	p-value
F1	600.3185322	9.06E-121
F3	586.9128153	6.60E-118
F4	618.7111078	1.07E-124
F5	602.046269	3.88E-121
F6	597.9813895	2.86E-120
F7	598.7280879	1.98E-120
F8	616.4663436	3.22E-124
F9	600.7256889	7.42E-121
F13	624.8521268	5.21E-126
F14	605.2327044	8.09E-122
F15	614.0220801	1.07E-123

	$\tilde{\chi}^2$	p-value
F16	609.9777441	7.84E-123
F18	627.0513478	1.76E-126
F21	609.9917873	7.79E-123
F23	599.8078185	1.17E-120
F24	620.2379775	5.04E-125
F25	617.8855294	1.60E-124
F44	619.976097	5.73E-125
F26	609.5763896	9.55E-123
F45	602.0403012	3.89E-121
F22	601.765704	4.45E-121
F46	562.0099834	1.36E-112

Moreover, we can observe the significant difference in Table- 17. GSA algorithm win on twelve functions, however, METO secures second place on sixteen functions. From all algorithms, we can observe that on average METO is the best algorithm. CMSA algorithm gives the best solution on six functions but no other functions, and it has the worst performance on eleven functions as it is on the 13^{th} position. We highlight all good results in bold in the tables. We can observe that on most of the funcitons METO secures second position where METO is not very good but better than the others. It motivates us to improve it by tuning its parameters, which we left open for future research. $\mathscr{C}$ of METO is significant comparatively.

We can observe that where METO is not best on functions, the average performance is comparative to validate the ability of METO to be a substitute optimizer for most of the functions. Significance difference between the distribution of METO and other optimizers can be observed in the box-plot of Fig. 15. First box-plot belongs to the METO.

5.5 Limitation of METO

As "No free lunch" theorem [9] suggest that elevated performance of any optimizer over one class of problem costs on the performance over another type, METO holds the same. From the observation, we can draw the limitations of METO, where it underperform on plate-shaped and bowl-shaped functions on the higher dimension. We observed that plate-shaped functions such as Zakharov function, Perm0db, etc. , cannot be solved by METO. Performance of METO is not very efficient for the class of functions where more than one global points exits, such as Griewank, Weierstrass, etc. functions. From the experimentation on the vast set of problems, we observed that best performance from METO could not be expected for all class of problems. By following the results, we found that for few functions such as Ackley, Rosenbrock Leon, Dixon price, CF-3, HCF-1 optimizer CMAES, GSA, SLPSO, BBO, and TLBO give the better solution than METO. In the case of these function by tuning the parameters of METO near solution can be found. By observing the results, one can conclude that METO surpasses other optimizers on multi-modal with a single global solution, non-linear, steep/ridge, flat surface/step integer problems, and discrete problems. However, for the bowl-shaped like sphere function, it gives comparative results but requires the number of function evaluations.

5.6 Parameter optimization

Parameters of the METO are problem-specific. Using the sensitivity analysis by utilizing multiple Monte Carlo runs on a particular problem, we can define a range of all parameters for better performance. The proposed algorithm is sensitive to the following parameters, tuning of which can improve the solution accuracy.

1 - Number of bits in the DNA to represent the variable. As we have discussed in Section 3.1 that the accuracy of the solution can be increased by higher number of N_b. But it increases the size of chromosome, thus extra computational burden for higher number of variables. Based on the required computational accuracy, it can be increased.
2 - Lower and upper limits of mutation probability τ is defined by equation 30. In initial iterations, it is high to explore maximum search space, which is reduced exponentially to very low value to fine-tune the solution in later evolution epochs.
3 - Flipping probability δ is an important factor and defined in equation 14. Since it fillips the SS and produces completely different AS of the DNA, thus, it needs to be defined carefully. The high value of it avoids the local extreme in the initial phase of evolution but affects the later stage of evolution when fine-tuning of the solution is required. Thus, in the later stage of evolution iteration, it should be lower enough. However, for a large number of variables, instead of equation 14, τ can be generated randomly in the range [0, 0.3] for each selected DNA of the chromosome from fist iteration to the last. It may provide an efficient search. The optimal value of it needs rigorous experimentation, which is our future research scope.
4 - Cross-breeding and self-breeding and Epimutation rates are again important factors. Appropriate selection of them reduced the number of function evaluation. The value of them should not be small as well as large. In our proposal we recommends its value in Algorithm-5, by Δ_{F1} and Δ_{F2}. Epimutation rates should be the same as Δ_{F2}.
5 - Smaller population size may give quicker convergence, but there is a risk to be trapped at a local extreme. Vice-e-verse, large population size can lead to process speed down. Similarly, a low number of bits N_v tends for lower

Table 16: Kruskal Wallis Rank Sum Score: 100 variables

	METO	BHGA	BBO	IWO	DE	CMAES	SFLA	FA	TLBO	CUCKOO	BA	GSA	SLPSO
F1	36.9	158.8	180.8	301.4	522.1	625.4	575.6	419.9	259.4	419.5	402.3	65.0	264.3
F3	76.5	182.6	237.5	363.2	463.8	25.5	574.7	431.1	426.1	331.2	364.7	625.2	129.4
F4	25.6	142.9	175.3	268.1	396.7	624.5	574.0	418.7	446.2	319.1	243.2	75.7	521.7
F5	38.6	290.3	225.9	372.7	436.3	135.7	625.5	545.1	134.0	550.7	336.9	113.9	425.9
F6	37.8	155.1	267.8	252.8	470.8	434.1	568.2	351.6	137.3	417.1	445.4	620.3	73.1
F7	81.9	352.6	210.0	298.8	495.3	96.8	625.5	499.2	314.6	574.5	416.9	164.9	100.7
F8	25.5	95.2	131.3	212.4	409.0	596.7	581.2	488.1	534.1	361.8	310.2	171.5	314.5
F9	25.5	103.3	125.9	214.4	303.2	596.5	600.7	465.2	514.7	362.0	358.5	179.0	382.6
F13	75.5	125.5	197.4	388.0	296.8	613.7	586.3	452.8	234.8	447.8	277.0	25.5	510.6
F14	75.5	131.4	226.8	475.6	295.7	599.0	565.4	391.0	224.6	514.9	255.0	25.5	451.3
F15	29.1	129.7	171.3	482.7	230.5	620.3	526.2	392.5	416.3	362.3	271.2	71.9	527.5
F16	66.5	216.3	134.7	349.4	453.6	615.7	573.0	411.3	534.3	345.0	296.0	34.7	200.9
F18	75.5	320.1	187.5	508.5	218.6	25.5	625.5	491.4	271.0	571.0	360.3	431.2	145.7
F21	37.6	162.5	202.1	487.0	292.4	610.3	584.6	418.1	314.7	468.5	150.3	79.5	423.9
F23	71.7	280.7	221.5	334.1	458.3	131.9	625.5	489.9	158.1	510.8	389.1	53.0	506.9
F24	87.8	223.0	150.1	236.0	474.1	625.3	575.7	382.5	519.5	385.5	365.9	25.5	180.6
F25	55.3	323.3	212.8	378.0	558.2	146.9	625.5	505.8	201.3	456.5	474.7	47.0	246.1
F44	25.5	235.5	178.5	342.4	543.5	113.4	625.5	508.8	359.0	449.4	499.5	107.3	243.2
F26	77.7	127.5	205.6	425.1	338.9	623.9	562.9	421.6	241.8	461.1	232.7	25.5	487.4
F45	31.7	151.8	264.7	254.2	457.5	472.1	562.9	342.1	143.1	412.0	443.7	620.0	75.8
F22	74.9	125.5	452.1	541.0	287.6	26.1	625.5	400.3	299.6	448.8	193.8	507.9	248.5
F46	76.2	304.1	152.0	267.2	507.8	452.2	609.7	463.7	503.5	402.2	292.8	173.5	26.6

Table 17: Rank of all optimizer based on Kruskal Wallis Rank Sum score: 100 variables

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}	TF	ω	Average-ω	Winner
METO	**11**	**11**	0	0	0	0	0	0	0	0	0	0	0	22	**275**	**21.2**	**1**
BHGA	0	2	7	4	3	2	3	1	0	0	0	0	0	22	211	16.2	12
BBO	0	0	5	7	7	2	0	0	0	1	0	0	0	22	208	16.0	2
IWO	0	0	0	0	5	3	4	4	1	2	2	1	0	22	142	10.9	3
DE	0	0	0	0	2	3	3	1	2	5	3	3	0	22	114	8.8	13
CMAES	3	1	3	1	0	0	0	0	2	0	1	1	10	22	119	9.2	11
SFLA	0	0	0	0	0	0	0	0	0	0	1	12	9	22	36	2.8	9
FA	0	0	0	0	0	0	2	6	2	8	4	0	0	22	104	8.0	4
TLBO	0	0	3	3	2	2	3	1	2	1	5	0	0	22	153	11.8	6
CUCKOO	0	0	0	0	0	1	3	5	6	2	1	4	0	22	108	8.3	5
BA	0	0	1	1	2	6	3	4	2	3	0	0	0	22	154	11.8	10
GSA	7	6	0	4	0	0	0	0	1	0	1	0	3	22	214	16.5	8
SLPSO	1	2	3	2	1	3	1	0	4	0	4	1	0	22	164	12.6	7
MF	13	12	11	10	9	8	7	6	5	4	3	2	1				

accuracy but provides faster computation. Thus, an appropriate selection of it can increase the computation power of the algorithm. An optimized trade-off between the number of species and its size may lead to better results, which is problem specific. Instead of increasing the population size increasing number of species gives considerably better result for large problem space.

6 - As the discussion given in -4, it is expected that best genes should dominate in offspring. Simulation results and effect of different values of ρ are shown in Fig. 14. Based on the experiments, we proposed it varying between the range 0.9 to 0.97.

7 - Epimutation factor ξ is also problem-dependent. It increases the function evaluations. A large number of ξ may give extra burden on the computation time but could improve the convergence as shown in Fig. 14. Thus it needs attention to define it carefully. For large variables problem space it can be increased to five.

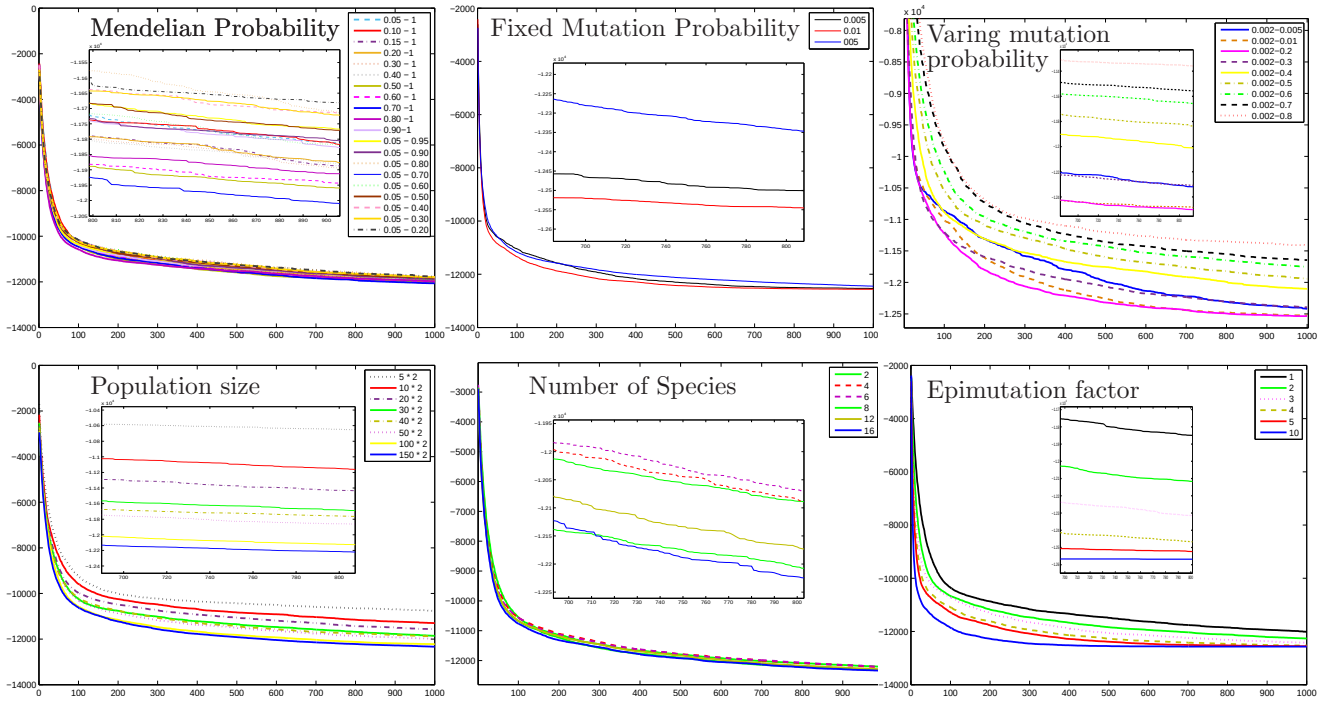


Fig. 14: Effect of parameter tuning

Another parameter may be the probability of transferring dominant genes from both parents to produce F1 generation offspring. We simulated it for 0.5, which follows the theory that half-half genes come from each parent plants. Although, dynamic selection of this parameter may play a significant role in global exploration, where the parent with high fitness may be more dominating than the other parent. The analysis of it is our future research scope.

6 Future research scope

It is worth to discuss the possible improvements in the basic structure of the METO. Pollination scheme stands at the First stage of improvement, where to introduce more diversity for the solution, random pollination, tournament pollination, roulette wheel pollination, etc. and hybrid technology of them can be tested. In the proposed optimizer, offspring generation from pure-bred plants contained in the same species is not considered, which may be a critical decision factor and thus future research scope. An adaptive mechanism for ρ, τ and ξ may be an impacting factor to improve the efficiency of the proposed optimizer. The choice of optimal parameters can be extracted using some methods for example fuzzy approach and machine learning techniques like the neural network.

After analysis of the above parameters, a question arises here, which individual should go to the next evolution? Different selection schemes can be utilized here, such as random, roulette wheel, tournament, reward-based, stochastic universal sampling, proportionate fitness selection, etc. Proposed algorithm performance with the different selection schemes is left open for future research. Also, mutation of the resulting offspring of $F1$ and $F2$ generation may improve the performance of the optimizer, which is out of the scope of this paper and could be analyzed in the future. Moreover, by utilizing the methods as given in [58, 59], binary tree memory, and spline interpolation, the computational time to evaluate functions for the same chromosomes can be saved. As a result, the faster algorithm can handle huge variables problems like training of deep neural network with millions of parameters very efficiently.

Hybridization of METO with other meta-heuristic algorithms [60] and Fuzzy logic computation [61] may be a better strategy to solve plate-shaped functions. At this juncture, we can observe that METO has a broad scope of further improvement and maybe a substitute optimization algorithm of GA and other optimizers for most of the large-scale complex problems from sociology, engineering, networks, topology, graph, biology, etc. Additionally, METO has one benefit of hardware friendly algorithm due to its processing in the binary coded system. Thus, a proposal of developing associated hardware to be implemented with embedded and VLSI systems is left open as a future development.

7 Conclusion

This paper presents a novel bio-inspired meta-heuristic binary coded optimization technique. Its algorithm imitates the Mendelian evolutionary theory proposed by G. Mendel

on multiple species of plants, where the recessive genes are transmitted to next generation with some probability. Thus, based on the evolution theory, five operators as called the Flipper, the Pollination, the Breeding, the Discrimination, the Epimutation are introduced and sequentially employed.

The algorithm imitates a two-strand DNA structure rather than a single chromosome one, which is formed by fertilization of sense strand of one plant DNA with the anti-sense strand of another plant DNA. By assigning a recessive DNA corresponding to each plant as the pseudo-global point, the algorithm does not deviate from optimal. A member with the best surviving value associated with a plant is assigned to as recessive chromosome and focuses on the exploration of neighboring points based on Mendelian transfer probability of genes. To benchmark the proposed optimizer, we have compared it with thirteen best-known optimizers of different nature, as well as a diverse set of test problems. Simulation results and statistical analysis on thirty and hundred variables test problems ranks the METO higher than the other optimizers. From the presented results, we can observe the better consistency of METO on multi-modal and steep-ridge, noisy and deceptive in nature functions. ACCs show clear-cut distinction between the algorithms on various types of test functions. Although, the limitation of the METO on plate-shaped, bowl-shaped problems is observable, where BBO, TLBO, RSA. In conclusion, METO is a nature-inspired genetic evolutionary algorithm which utilizes the standard structure of genotype and phenotypes for a double-strand DNA. Thus, it can be used on the broad range of the problems without any specific guidelines. Future works are expected to be done on the parametric control, complexity analysis, convergence analysis, reducing the number of functions evaluation, testing the algorithm in solving real-life problems, hardware implementation and extension for the multi-objective optimization problem case.

Acknowledgements We would like to thanks, Prof. L. M. Chamra, Dean of School of Engineering and Computer Science, Oakland University and Prof. Noboru Babaguchi for their support and provided facilities to make the proposed research work possible.

Compliance with ethical standards Conflict of interest The authors declare that they have no conflict of interest. Ethical standard This article does not contain any studies with human participants or animals performed by any of the authors.

8 Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest. **Ethical standard** This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. M. Khosravy, N. Gupta, N. Patel and T. Senjyu, “Frontier Applications of Nature Inspired Computation, Springer”, 2020.
2. N. Dey, A. Ashour, S. Bhattacharyya, “Applied Nature-Inspired Computing: Algorithms and Case Studies.” Springer, 2020.
3. BoussaïD, Ilhem, Julien Lepagnot, and Patrick Siarry. ”A survey on optimization metaheuristics.” Information sciences 237 (2013): 82-117.
4. Paszkowicz, Wojciech. ”Genetic algorithms, a nature-inspired tool: a survey of applications in materials science and related fields: part II.” Materials and Manufacturing Processes 28, no. 7 (2013): 708-725.
5. Liang, Y., Wang, L. Applying genetic algorithm and ant colony optimization algorithm into marine investigation path planning model. Soft Comput (2019), doi:10.1007/s00500-019-04414-4.
6. Tang, Deyu, Zhen Liu, Jin Yang, and Jie Zhao. ”Memetic frog leaping algorithm for global optimization.” Soft Computing 23, no. 21 (2019): 11077-11105.
7. AlRashidi, Mohammed R., and Mohamed E. El-Hawary. ”A survey of particle swarm optimization applications in electric power systems.” IEEE transactions on evolutionary computation 13, no. 4 (2008): 913-918.
8. Yuen, Shiu Yin, Yang Lou, and Xin Zhang. ”Selecting evolutionary algorithms for black box design optimization problems.” Soft Computing 23, no. 15 (2019): 6511-6531.
9. Wolpert, David H., and William G. Macready. ”No free lunch theorems for optimization.” IEEE transactions on evolutionary computation 1, no. 1 (1997): 67-82.
10. Parejo, José Antonio, Antonio Ruiz-Cortés, Sebastián Lozano, and Pablo Fernandez. ”Metaheuristic optimization frameworks: a survey and benchmarking.” Soft Computing 16, no. 3 (2012): 527-561.
11. Igel, Christian. ”No free lunch theorems: Limitations and perspectives of metaheuristics.” In Theory and principled methods for the design of metaheuristics, pp. 1-23. Springer, Berlin, Heidelberg, 2014.
12. Gandomi, Amir H., and Xin-She Yang. ”Chaotic bat algorithm.” Journal of Computational Science 5, no. 2 (2014): 224-232.
13. Silberholz, John, Andrea Raiconi, Raffaele Cerulli, Monica Gentili, B. Golden, and S. Chen. ”Comparison of heuristics for the colourful travelling salesman problem.” International Journal of Metaheuristics 2, no. 2 (2013): 141-173.
14. Beheshti, Zahra, and Siti Mariyam Hj Shamsuddin. ”A review of population-based meta-heuristic algorithms.” Int. J. Adv. Soft Comput. Appl 5, no. 1 (2013): 1-35.
15. Holland, John Henry. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT press, 1992.
16. Khari, Manju, Anunay Sinha, Elena Verdu, and Ruben González Crespo. ”Performance analysis of six meta-heuristic algorithms over automated test suite generation for path coverage-based optimization.” Soft Computing (2019): 1-18.
17. Wang, Dongshu, Dapei Tan, and Lei Liu. ”Particle swarm optimization algorithm: an overview.” Soft Computing 22, no. 2 (2018): 387-408.
18. Bruno, Giuseppe, Raffaele Cerulli, and Monica Gentili. ”Models, algorithms and applications for location problems.” (2016): 871-873.
19. Eusuff, Muzaffar, Kevin Lansey, and Fayzul Pasha. ”Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization.” Engineering optimization 38, no. 2 (2006): 129-154.
20. Gupta, Neeraj, Mahdi Khosravy, Nilesh Patel, and Tomonobu Senjyu. ”A bi-level evolutionary optimization for coordinated transmission expansion planning.” IEEE Access 6 (2018): 48455-48477.

21. Simon, Dan. "Biogeography-based optimization." IEEE transactions on evolutionary computation 12, no. 6 (2008): 702-713.
22. Rajabioun, Ramin. "Cuckoo optimization algorithm." Applied soft computing 11, no. 8 (2011): 5508-5518.
23. Yang, Xin-She, "Bat algorithm for multi-objective optimization," International Journal of Bio-Inspired Computation, vol. 3, no. 5 (2011): 267–274.
24. Rao, R. Venkata, Vimal J. Savsani, and D. P. Vakharia. "Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems." Computer-Aided Design 43, no. 3 (2011): 303-315.
25. Chen, Xianshun, Yew-Soon Ong, Meng-Hiot Lim, and Kay Chen Tan. "A multi-facet survey on memetic computation." IEEE Transactions on Evolutionary Computation 15, no. 5 (2011): 591-607.
26. Qin, A. Kai, Vicky Ling Huang, and Ponnuthurai N. Suganthan. "Differential evolution algorithm with strategy adaptation for global numerical optimization." IEEE transactions on Evolutionary Computation 13, no. 2 (2008): 398-417.
27. Das, Swagatam, and Ponnuthurai Nagaratnam Suganthan. "Differential evolution: A survey of the state-of-the-art." IEEE transactions on evolutionary computation 15, no. 1 (2010): 4-31.
28. Franciosi, Chiara, Francesco Carrabs, Raffaele Cerulli, and Salvatore Miranda. "An evolutionary approach for the offsetting inventory cycle problem." Cogent Engineering 4, no. 1 (2017): 1370764.
29. Patnaik, S N., Rula M. Coroneos, James D. Guptill, and Dale A. Hopkins. "Comparative evaluation of different optimization algorithms for structural design applications." International journal for numerical methods in engineering 39, no. 10 (1996): 1761-1774.
30. Agbolosu-Amison, Seli James, Byungkyu Park, and Ilsoo Yun. "Comparative evaluation of heuristic optimization methods in urban arterial network optimization." In 2009 12th International IEEE Conference on Intelligent Transportation Systems, pp. 1-6. IEEE, 2009.
31. Cerulli, Raffaele, Andreas Fink, Monica Gentili, and Stefan Voß. "Metaheuristics comparison for the minimum labelling spanning tree problem." In The Next Wave in Computing, Optimization, and Decision Technologies, pp. 93-106. Springer, Boston, MA, 2005.
32. Martins, Lilian Al-Chueyr Pereira. "The dissemination of the chromosome theory of Mendelian heredity by Morgan and his collaborators around 1915: a case study on the distortion of science by scientists." Filosofia e História da Biologia 5, no. 2 (2010): 327-367.
33. Sinden, Richard R. DNA structure and function. Elsevier, 2012.
34. Frankel, Rafael, and Esra Galun. Pollination mechanisms, reproduction and plant breeding. Vol. 2. Springer Science & Business Media, 2012.
35. Oey, H., and E. Whitelaw. "On the meaning of the word 'epimutation'." Trends in genetics 30, no. 12 (2014): 519-520.
36. Higgs, Eric. Nature by design: people, natural process, and ecological restoration. MIT Press, 2003.
37. Power Systems and Evolutionary Algorithms [On-line] http://alroomi.org/benchmarks/unconstrained/n-dimensions.
38. Benchmarks for Evaluation of Evolutionary Algorithms [On-line] http://www.ntu.edu.sg/home/epnsugan/
39. S. Surjanovic, D. Bingham, "Virtual library of simulation experiments: test functions and datasets," Simon Fraser University, in: http://www. sfu. ca/ ssurjano/optimization. html [Links]. 2013.
40. Gupta, Neeraj, Nilesh Patel, Bhupendra Nath Tiwari, and Mahdi Khosravy. "Genetic algorithm based on enhanced selection and log-scaled mutation technique." In Proceedings of the Future Technologies Conference, pp. 730-748. Springer, Cham, 2018.
41. B. Xing, and W.J. Gao, "Invasive weed optimization algorithm," *In Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms,Springer, Cham,* (pp. 177-181). 2014.
42. Yang, Xin-She, and Suash Deb. "Engineering optimisation by cuckoo search." arXiv preprint arXiv:1005.2908 (2010).
43. Meng, Xian-Bing, Xiao Zhi Gao, Yu Liu, and Hengzhen Zhang. "A novel bat algorithm with habitat selection and Doppler effect in echoes for optimization." Expert Systems with Applications 42, no. 17-18 (2015): 6350-6364.
44. Yılmaz, Selim, and Ecir U. Küçüksille. "A new modification approach on bat algorithm for solving optimization problems." Applied Soft Computing 28 (2015): 259-275.
45. Rashedi, Esmat, Hossein Nezamabadi-Pour, and Saeid Saryazdi. "GSA: a gravitational search algorithm." Information sciences 179, no. 13 (2009): 2232-2248.
46. Cheng, Ran, and Yaochu Jin. "A social learning particle swarm optimization algorithm for scalable optimization." Information Sciences 291 (2015): 43-60.
47. Yang, Xin-She. "Firefly algorithm, Levy flights and global optimization." In Research and development in intelligent systems XXVI, pp. 209-218. Springer, London, 2010.
48. Hansen, Nikolaus, Sibylle D. Müller, and Petros Koumoutsakos. "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES)." Evolutionary computation 11, no. 1 (2003): 1-18.
49. Hansen, Nikolaus. "The CMA evolution strategy: a comparing review." In Towards a new evolutionary computation, pp. 75-102. Springer, Berlin, Heidelberg, 2006.
50. Islam, Sk Minhazul, Swagatam Das, Saurav Ghosh, Subhrajit Roy, and Ponnuthurai Nagaratnam Suganthan. "An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization." IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 42, no. 2 (2011): 482-500.
51. Nai, Corrado, Boris Magrini, and Julia Offe. "Let microorganisms do the talking, let us talk more about microorganisms." Fungal biology and biotechnology 3, no. 1 (2016): 5.
52. Stebbins, G. Ledyard. "Self fertilization and population variability in the higher plants." The American Naturalist 91, no. 861 (1957): 337-354.
53. Soremekun, G., Z. Gürdal, R. T. Haftka, and L. T. Watson. "Composite laminate design optimization by genetic algorithm with generalized elitist selection." Computers structures 79, no. 2 (2001): 131-143.
54. Lewontin, Richard C. The genetic basis of evolutionary change. Vol. 560. New York: Columbia University Press, 1974.
55. Calo, Silvia, Cecelia Shertz-Wall, Soo Chan Lee, Robert J. Bastidas, Francisco E. Nicolás, Joshua A. Granek, Piotr Mieczkowski et al. "Antifungal drug resistance evoked via RNAi-dependent epimutations." Nature 513, no. 7519 (2014): 555.
56. Chan, Yvonne, and Roy P. Walmsley. "Learning and understanding the Kruskal-Wallis one-way analysis-of-variance-by-ranks test for differences among three or more independent groups." Physical therapy 77, no. 12 (1997): 1755-1761.
57. Kruskal Wallis Test in Matlab [online]: https://in.mathworks.com/help/stats/kruskalwallis.html
58. Yang, Shengxiang. "Genetic algorithms with memory-and elitism-based immigrants in dynamic environments." Evolutionary Computation 16, no. 3 (2008): 385-416.
59. Gantovnik, Vladimir B., Zafer Gürdal, and Layne T. Watson. "A genetic algorithm with memory for optimal design of laminated sandwich composite panels." Composite Structures 58, no. 4 (2002): 513-520.
60. Cuevas, Erik, Alma Rodríguez, Arturo Valdivia, Daniel Zaldívar, and Marco Pérez. "A hybrid evolutionary approach based on the invasive weed optimization and estimation distribution algorithms." Soft Computing 23, no. 24 (2019): 13627-13668.
61. Di Nola, Antonio, Ada Lettieri, Irina Perfilieva, and Vilém Novák. "Algebraic analysis of fuzzy systems." Fuzzy Sets and Systems 158, no. 1 (2007): 1-22.