

Research Article

Modified Environmental Adaptation Method and its Application in Test Case GenerationShailesh Tiwari^{1,*}, Anuj Kumar², Akash Punhani³ and K. K. Mishra⁴¹ABES Engineering College, Ghaziabad, UP, India²IEC-CET, Greater Noida, UP, India³Amity University, Noida, UP, India⁴MNNIT Allahabad, Prayagraj, UP, India

Received 14 October 2019; Accepted 29 December 2019

Abstract

A modified environmental adaptation method (MEAM) is the modified version of environmental adaptation method. The MEAM algorithm is operated based on adaptation and alteration operator to find the solution from the huge search space. The alteration operator in MEAM is used to evade the incidence of being stuck into local minima. To test the performance of on real world application MEAM is applied in test case generation problem and proved to be better than PSO-TVAC algorithm. From the results it has been observed that there is the improvement of 4% percent that the PSO-TVAC.

Keywords: Environmental Adaptation method, Particle swarm optimization (PSO), Regression testing, Test case generation

1. Introduction

In Optimization, the exploration of the best solution is performed from a large set of solutions to the specific problem. The identification of the best solution is done with the help of fitness function also termed as objective function. The optimal solution to the problem is either maximal or minimal value of the fitness function. Thus, optimization algorithm can be defined as an algorithm that searches for the optimal solution in the target search space termed as problem search space. Based on the problem this search space can either be continuous or discrete in nature. Every value on the search space represents one solution.

The optimization algorithm can be deterministic or randomized based on the complexity of the search problem. For the small number of points in the search space, the basic algorithms based on dynamic programming can be used, to identify the optimal solution by evaluating the fitness function for all possible points. However, for the problems with a large number of solutions, the solution using dynamic programming would be infeasible. These problems come in the category of NP-hard and it will be better to evaluate nearly optimal solutions or local optimal solution. Local solution can be identified by either of the two ways: By finding the derivative of the function defining the problem to be minimized or maximized. The algorithms that employ the derivative calculation are called as a gradient-based algorithm. On the other hand, the solution based on heuristics and follow no definite process are known as randomized algorithms. The randomized algorithm is helpful in finding the solution for NP-hard problems but has the drawback that the solution is near to the optimal solution and may not be the perfect optimal solution. The widely used randomized

algorithms are evolutionary programming, evolutionary strategy, genetic algorithm, particle swarm optimization, and genetic programming.

Even though there is huge literature available for solving optimization problems, nevertheless all these algorithms have the same objectives. These objectives may be

- Either to minimize the total number of generations required to achieve the optimal solution.
- The algorithm must be clever to avoid the local minima or maxima and identify different optimal solutions for multi modal problem.

This paper deliberates various design objectives that are to be focused on optimization algorithms and proposes an algorithm for optimization.

1.1 Convergence Rate

As the main criteria for the better optimization algorithm are to reduce the number of fitness function evaluations required to reach the optimal solutions. The number of function evaluations can be reduced if the algorithm is high convergence rate. Convergence rate actually defines how quickly the algorithm identifies the optimal solution. In order to attain high convergence, it is necessary to know the parameter on which the convergence rate is dependent. On the exhaustive review of the randomized algorithm, it has been identified that the convergence rate depends upon the mapping of the problem to natural phenomena that is used.

Randomized algorithms are designed with an objective of solving the complex problem of optimization. As a result, the randomized algorithm will contain a set of steps that are executed to explore the solution. However, the user is not aware about the properties of the objective function. Hence, there is a need of mapping between algorithm and function that can guide the search. The mapping of natural phenomena with the problem has provided a tool for the exploration of the

*E-mail address: shailesh.tiwari@yahoo.com

ISSN: 1791-2377 © 2019 School of Science, IITU. All rights reserved.

doi:10.25103/jestr.126.26

optimal solution when the user is not aware of the search direction. Due to this, many randomized algorithms are inspired by nature and their searching power is based upon the mapping of problem with the natural phenomena. In past, many nature-inspired algorithms have been proposed but still, there is scope for new optimization algorithms that can explore more quickly.

1.2 Ability to Capture Multiple Optimal Solutions

As the single objective optimization algorithms have two variants that are unimodal or multimodal. In the case of the unimodal problem, there is only one possible optimal solution but in case of multimodal optimization, there is more than one global solution. To identify multiple global solutions it is essential to explore the complete search space until all possible solutions are not obtained. To do so the randomized algorithms have the predefined set of operators that can explore and exploit the search space efficiently. There are some control parameters associated with the algorithm that can keep a check on the exploration.

Unlike in Unimodal optimization, in which the probable area is explored once the area is identified the exploitation of this area is done. However, in multimodal optimization, as there is the possibility of finding more than one area with the optimal solutions hence the exploration algorithm can stop as it is not certain that the area explored has the global optimal solution. Therefore, the key difference between the unimodal optimization algorithm and the multimodal optimization algorithm is that the exploration may stop after few initial steps but in the case of a multimodal optimization algorithm there must be exhaustive exploration and exploitation.

Most of the existing randomized algorithms are designed to find a single optimal solution like GA Holland [20], DE and PSO. These algorithms were best suited for unimodal algorithms than for multimodal optimization algorithms. However, these algorithms can be modified for multimodal optimization problem [1].

The example in the case of a simple genetic algorithm except for mutation the remaining operations work for the exploitation of the identified sub section of the problem area [2][21]. As there is very little emphasis on exploration of the search space this leads the genetic algorithm to stuck into local minima [1]. In order to modify the simple genetic algorithm for multimodal problems, the sharing function theory was introduced in the genetic algorithm. In the case of a Differential evolutionary algorithm, there is a need to modify the crossover and differential operators with the target to capture solution for multi-modal problems. Particle swarm optimization also requires the turning of the parameters to identify the solution for multimodal solutions [3, 4]). The parameter tuning in most of the unimodal algorithms can help in achieving the solution from the algorithm such that proper weightage to both the exploration of new areas and exploitation of the existing areas identified by the exploration module.

1.3 Appropriate Adjustment of Random Parameters

Most of the randomized algorithms are based on nature and there are some parameters that help in the random exploration and exploitation in the search space. The different algorithms have different parameters like genetic algorithm have the parameters like probability of crossover P_c and Probability of Mutation P_m . These can vary the exploitation and exploration respectively [5]). Similarly, DE has parameters like F , CR [6, 7] and PSO have three parameters that are w, c_1 and c_2 [8, 9]. If these parameters are selected efficiently then the

convergence rate of the algorithm can improve significantly. Another advantage of proper tuning of parameter is that this will avoid the algorithm to get stuck into local minima. However, the tuning of the parameter of these algorithms requires the huge expertise and lack of which may lead to the poor performance of the algorithm. This problem can be eliminated if the auto-tuning of parameters can be done based upon the learning.

Consequently, to design a novel algorithm to generate multiple solutions for the multimodal problem that can tune on its own to achieve a high convergence rate without getting stuck into local minima.

In this paper, an optimization algorithm termed as Environmental Adaption Method (EAM) is presented. EAM has the target has three targets at the design time:

- To minimize computation time required for solving the problem.
- To get the multimodal solutions from the algorithm
- Algorithm should be capable of escaping from local minima.

In the present paper, a variant of environmental adaptation method has been proposed. The newly designed algorithm is compared the algorithm on benchmark functions. Moreover, the algorithm is applied to generate the test cases for white box testing. Before moving further, we will like to discuss in the background of differential evolutionary algorithm

2. Background Details

Nature has itself set the examples to solve the many real world problems in the proficient and operative manner. This has inspired most of the optimization algorithms for example: addition of strength to the metal by annealing process leads to the simulated annealing algorithm. The Darwin theory of natural selection has inspired a large number of algorithms that are termed as Evolutionary algorithms. These algorithms have been further studied with various intelligence used by the nature like Particle swarm optimization (PSO), Ant colony optimization (ACO), Artificial Bee colony optimization ABC [10, 11]. The search by the cuckoo bird has led to the development of Cuckoo algorithm and flashing behavior of fireflies has been inspired in the generation of Firefly algorithm.

Civicioglu et al. [22] have proposed to employ quantum computing in Evolutionary algorithms. [12–15] have highlighted the various application of the different Evolutionary algorithms. The conventional algorithms are mixed with evolutionary algorithms like cuckoo search in the enhancement of biomedical imaging and the segmentation of image based on an evolutionary algorithm has been suggested by Chatterjee et al. [23]. The Evolutionary algorithms can be blended with neural networks and other techniques as per the requirement of the problem. The similar idea was used by Chatterjee et al. [23] for the classification of dengue data using the cuckoo search and artificial neural networks.

The novel algorithms for optimization can be generated by employing the idea of adaptive learning. The idea of adaptive learning has not been employed in optimization problems as per the kinds of literature studied until now.

Let the population is transferred to the new environmental condition. These changes in the environment of the individual

will induce the changes in the plastic traits generally referred as the conduct of the individual. These changes are due to the changes in the genetic structure of the individual. The genetic structure stores the inherited data from the previous generation. However, there is the possibility of altering the gene structure at the time of reproduction between the individuals. The properties like the conduct of the person, their-working style may vary according to the environment. The changes in the genes due to the environment may be positive or negative for the survival of the individual. The positive changes in the genes support in the survival of the individual while the negative changes in the individual may lead to the challenges for the survival of the individuals. Now according to Darwin's theory of natural selection the positive changes related to the high fitness values and become part of the next generation.

M. Wund [24] has explained the process as "under novel environmental conditions, some changes in behavior, physiology, life history, and/or morphology might be beneficial, allowing at least some individuals to persist until natural selection further enhances a population's mean fitness". As the concept stated is self-capable of generating the optimal solutions, therefore, the concept can use for designing the optimization algorithms.

The above-discussed concept has been used in designing the environmental adaptation method (EAM). Fig. 1 Graphical Representation of EAM is the pictorial representation of the EAM. In order to design the EAM the following assumption has been taken into account. The average fitness function of the generation under consideration has been termed as environmental fitness. The environmental fitness keeps the check, that the particular individual is suitable for the particular environment or not. The individuals that are above environmental fitness with respect to their fitness are favored by the current environment while remaining individuals with low fitness value are struggling for their existence in the current environment. With the minute change in the environment will act as the triggers the entire population to update the genetic structure. If the individual already exists in the environment these changes improve the fitness of the individual while the other will improve to attain the minimum fitness criteria.

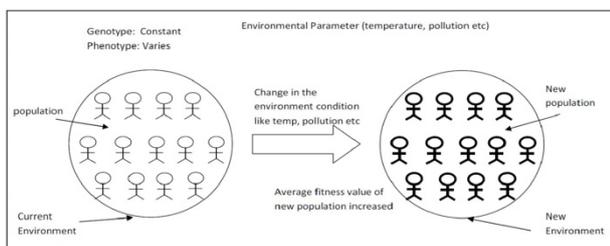


Fig. 1 Graphical Representation of EAM

With the objective of designing the Optimization algorithm, the three processes have been identified as the part of adaptive learning. These three processes are adaptation process, alteration process and selection process. The adaptation process is very deeply rooted in the alteration process. As from the above discussion, it is clear that change in the environmental setup will trigger the changes in the phenotypic structure of the initial population but as there is also the disturbance in the environment, which can be termed as noise. This noise in the environment can also affect involved in the changes of the individual of the populations. These changes due to the noise are termed as the alteration

operator. In the case of the algorithmic implementation of the population generated after the application of the alteration and adaptation operator is termed as an intermediate population. The individual of the intermediate population is operated with the selection operator, which selects the individuals with good solutions and the solutions that are not participating in the selection process are removed from the generation. These removed individuals may be considered as the dead. This, in turn, will increase the average fitness of the population and increases the probability of getting good solutions in a short span of time.

Let us consider the real life example of dengue mosquito to understand the concept of adaptive learning. Dengue mosquito belongs to the special class of mosquito named as "Aedes". This mosquito has developed by the modification in the phenotypic structure but an increase in the growth of the mosquito has posed danger to human survival. In past, these mosquitos have a low mortality rate due to the marginal influence of humans in the killings of the mosquitos. However, due to the modification of the environment with the use of pesticides in various forms like coils, vaporizers to reduce the population of mosquitos. As nature has inbuilt mechanism of adaptive learning the mosquitos have adapted to the new environment by changing their genetic structures. The speeds of these changes are recorded with in the two months in the change of environment.

3. Proposed Approach

EAM method is similar to GA and PSO and is based on the random population that is generated at the initial level. This randomly generated population is operated with the three sets of operators that are as follows. Initially, the adaptation operator is applied to the population. This operator is responsible for modifying the genetic structure of the individual based upon the current fitness value of the individual and the environmental fitness value of the population. Now to introduce the influence created by the environmental noise the alteration operator is used. This generated the intermediate population. Now at the last, the selection of the best solutions from the current and the intermediate generation is done with the assistance of the selection operator. These operators are executed iteratively unless the threshold is reached. Fig. 1 describes the complete working of Environmental Adaptation Method (EAM). Both EAM and EA (Evolutionary Algorithm) have both have similar procedures for generating the initial population. However, the main dissimilarity among the EA and EAM is that EA considers the two best solutions for generating the new solutions, on contrast EAM considers the average fitness value of the complete population to modify the genes of the newly generated solution. Another major difference between the EA and EAM is the way of applying mutation and its counterpart alteration operator. The mutation operator is simple and is flipping a single bit based on the probability but alteration operator is a complex operation and may influence the complete genes of the population.

EAM is likewise not quite the same as complex algorithms like Cuckoo search algorithm. In the case of EAM, the numbers of operations that are applied to every individual in the particular generation are finite and constant; the same can be verified from.

On other hand, the cuckoo search skips some of the generations based upon the rate of unearthing of alien eggs (Civicioglu et al. [34]).

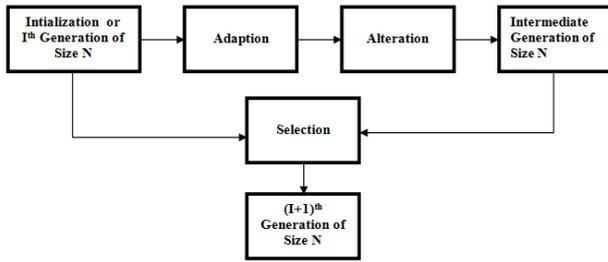


Fig. 1 Working of MEAM

Proposed Algorithm.

The algorithm for EAM is stated below

- Input: MAX_Gen (maximum number of generations)
- Pop_Size (Population size)
- Output: Q* (The final population representing the solution)
- Other variables: n (Counter that is used to count the number of generations)
- O' (transitional offspring)
- O (offspring generate after the application of operator of EAM)
- P (Set containing the population)
- in (ithsolution of nth generation)

- Step 1. Assign n ← 0.
- Step 2. Create random population POP₀ containing Pop_Size individuals.
- Step 3. Now on every individual in population apply adaption operator using the equation stated below to generate transitional population O'_n, in the equation described below P_{in} decimal transformation of binary representation of the individual.

$$\left(O'_{in} = \alpha * \left(\text{Decoded value in decimal of binary coding of } P_{in} \right)^{\frac{F_n(P_{in})}{F_{Avg}}} + \beta \right) \% (2^L) \tag{1}$$

- Step 4. Based on the probability decided to apply an alteration operator and generate P_n.

$$P_n = \{POP_n \cup O_n\}$$
- Step 5. Increase the generation count.
- Step 6. if n < MAX_GEN then go to Step 9 else goto Step 8
- Step 7. Q* ← best solutions(P_n)
- Step 8. Return Q*
- Step 9. POP_n = Pop_Size solutions with best objective value in P_{n-1}; goto Step 4

3.1 Details of the algorithm

Fig. 2 describes the flowchart of MEAM.

3.1.1 Forms of Encoding

As the proposed algorithm is capable of supporting multiple encoding techniques but the initial version of EAM was introduced with the binary encoding technique. The main advantage of using the binary encoding that is the degree of flexibility provided by this encoding technique on other hands if real coded parameters are used for the encoding then more updates are required in the algorithm based on the specific problem. The binary encoding problem divides the complete search space of the problem into fixed interval quantum. This helps in representing 2ⁿ solution if there are n discrete partitions that are created. Once the encoding process is over

and the solution has been identified using the algorithm the solution has to be decoded again. This decoding can be done by the easily using simple mathematical formula. However, in some cases, it is not feasible to convert the problem into the binary version. In such cases, it becomes essential to use a real encoding. The better part of the EAM is that it works equally well with binary as well as real encoding techniques.

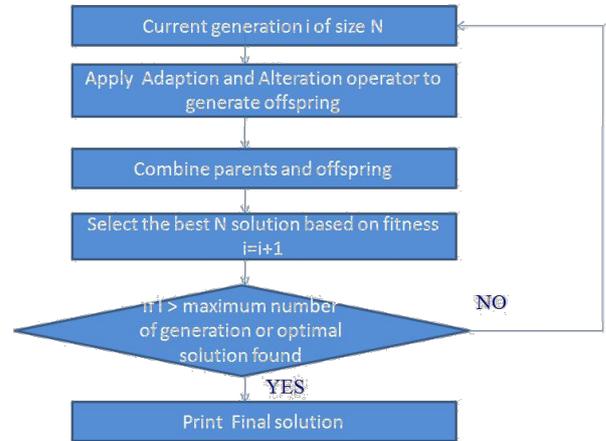


Fig. 2 Flowchart of the proposed algorithm

3.1.2 Population Initialization

The first step in the execution of the EAM is the generation of the random population with the specified number of individuals and is termed as the population size. The population size is identical to the population size of GA, PSO, DE. The procedure for selecting the optimal population size is the same as in other heuristic approaches mentioned above.

3.1.3 Creating Next Generation

The process of generating the new generation from the current generation is performed with the help of three operators that are an adaptation, alteration, and selection operators. The detailed function of the three operators is expounded one by one in this section.

3.1.4 Adaption Operator

It is the key operator of EAM algorithm. It ensures both the exploration of the problem search space as well as the exploitation of the search regions identified. The new offspring for the intermediate generation is generated by each individual and is described by the equation (2).

$$P_{i+1} = \alpha x \left((\text{Decoded value in decimal of binary coding of } P_i)^{\frac{F(P_i)}{F_{avg}}} + \beta \right) \% 2^L \tag{2}$$

In the above equation F(P_i) is a function that is used for the computation of the fitness value of the particular individual that is passed to the function. There are two control variables 'α' and 'β' that are specific to the problem under consideration. L is the number of bits that are used to represent the maximum value of the individual. The modulus operator helps in checking the value form exceeding the range defined in the problem. The exponential ratio of current fitness value to average fitness helps in creating the damping effect. However, when this value changes to one once the global maxima is reached. The damping effect is created by the exponential function and power will reduce to one when global is obtained. In the case of local minima, the 'β' will

help in escaping from the local minima as each iteration will add 'β' to the preceding value.

In order to automate the process of learning the value of 'α' and 'β' parameters are to be selected in such a way that algorithm always approaches the optimal solution.

The following values $\alpha = 2 \frac{I(F_{max} - F_{avg})}{F_{avg}}$ and

$$\beta = F_{max} - \frac{F(P_i)}{F_{max}}$$

are used for directing the search in the direction of optimal solutions. In the relation of for alpha and beta, 'F_{max}' represents the maximum fitness value in the particular generation. The reason behind defining alpha and beta in a particular manner can help in search of optimal solutions is as follows. During the starting generations, it is necessary to explore the complete search space which requires the value of 'α' to be high and should reduce slowly in late generations this is validated as the difference between 'F_{max}' and 'F_{avg}' is high in starting but in later generations both will tend toward to zero leading the value of 'α' to 1. When α is equal to one this means there will be randomness in the solutions. if 'α' = 1 and 'β' = zero, this means that solution points to the global optimal solution. It has been studied that the complexity of the algorithm is O(n).

3.1.5 Alteration Operator

The alteration operator is behaving with two different roles in EAM. In starting iteration of EAM, the alteration operator helps in exploiting the particular region of the problem search space for a good solution. However, in the later generations, it helps in the exploration. The algorithmic representations of the alteration operator are shown below:

Repeat for every individual in the population

Repeat for every bit I_m of an individual;

If P_{alt} < (rand()*100 - Pop_Gen*100/MaX_GEN)

then

Flip the bit

From the algorithm, it is clear that every individual of the current generation has to undergo the process of alteration and every bit of the is flipped with the probability value defined by the user. This is to ensure that diverse solutions are obtained and removes the chances that the algorithms stuck into the local minima. The complexity of the alteration operator is O(mn) where m is the number of bits and n is the population size. There are some modifications that changes

Table 1 Benchmark function with their type and Range

Function	Mathematical Representation	Range	Function Type
Rastrigrin	$f(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	[-10.0, 10.0]	Multimodal
Griewank	$f(x) = \sum_{i=1}^n \frac{x_i^2}{4000} - \pi_i^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	[-600.0, 600.0]	Multimodal
Axis Parallel Hyper-Ellipsoid	$f(x) = \sum_{i=1}^{n-1} i x_i^2$	[-5.12, 5.12]	Unimodal
Sphere	$f(x) = \sum_{i=1}^n x_i^2$	[-100.0, 100.0]	Unimodal
Schwefel	$f(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	[-500, 500]	Multimodal
Quartic	$f(x) = \sum_{i=1}^n i x_i^4 + \text{Rand}[0, 1]$	[-1.28, 1.28]	Unimodal
Zakharov	$f(x) = \sum_{i=1}^n x_i^2 + (\sum_{i=1}^n 0.5 i x_i)^2 + (\sum_{i=1}^n 0.5 i x_i)^4$	[-1.28, 1.28]	Unimodal

In the above table, x* represents the best solution and the value obtained by the function is represented by f(x*). To analyze the performance all the three algorithms were executed on these benchmark functions defined in Table 1. The benchmark function was executed on three different dimensions that are 10, 20, 30. The two different population size was used to analyze the performance that is 50 and 100

the rate of alteration operator to the final stage of the executions as the role of exploration has changed to the exploitation of the area.

3.1.6 Selection

The intermediate population and the current population are merged together to generate the set of 2N solutions, where N is the population size. These 2N solutions are initially sorted based upon their fitness values and the N solution with the best fitness values are copied to the next generation and the remaining solution is discarded.

3.1.7 Generation Phase and Progression

After the generation of the new population generation, the fitness function is checked with the threshold fitness or the generation count is checked with the expected counts set by the user. If the conditions are satisfied then the algorithm is terminated otherwise another set of iteration is executed on the obtained generation.

The Proposed MEAM algorithm is compared with the two best available versions of DE and PSO. The best-known version of DE that is SADE and for PSO-TVAC is the variant of PSO used for the comparison of the results. The three algorithms are tested on the seven benchmark function out of which some are unimodal and remaining are multimodal functions. Table 1 describes the mathematical formulae for the seven-benchmark function along with their range and type of function.

3.2 Unimodal Function

For the study of unimodal functions, the three functions are considered in the study. The names of these functions are axis parallel hyper-ellipsoid function, sphere function, and quartic function. These functions are used to evaluate the performance of the randomized algorithm that is SADE, PSO-TVAC, EAM and the proposed MEAM.

3.2.1 Performance of Unimodal Benchmark Functions

The performance of MEAM is tested on 3 unimodal functions as stated in table 1.

Table 2 represents the optimal value of the problem by the optimal value it can be understood that the value represent the global maxima.

respectively. The number of generations was taken as 1000 for all the three algorithms under the study. The statistics of minimum, maximum, average and standard deviation were recorded.

Based on the type of benchmark functions, the complete study can be categorized into two parts that are unimodal and multimodal functions. If the function has single optima then

it is called an unimodal function while it will be termed as Multimodal functions if having many global or local optimum. Table 1 describes the ranges and mathematical formulation of the benchmark functions used and Table 2 provides the details about the global Minimum value.

3.2 Unimodal Function

For the study of unimodal functions, the three functions are considered in the study. The names of these functions are axis parallel hyper-ellipsoid function, sphere function, and quartic function. These functions are used to evaluate the performance of the randomized algorithm that is SADE, PSO-TVAC, EAM and the proposed MEAM.

3.2.1 Performance of Unimodal Benchmark Functions

The performance of MEAM is tested on 3 unimodal functions as stated in table 1.

Table 2 Describes the global minimum value and function value of benchmark functions

Function	Global Minimum	
	x^*	$f(x^*)$
Rastrigrin	0	0
Griewank	0	0
Axis Parallel Hyper-Ellipsoid	0	0
Sphere	0	0
Schwefel	420.9687	-418.9829
Quartic	0	0
Zakharov	0	0

A. Axis Parallel Hyper Ellipsoid Function

This function is having single minima therefore; it is a unimodal function, which is continuous and convex in nature. The axis parallel hyper ellipsoid resembles to Dejong’s function. The curve of the function is described in Fig. 3.

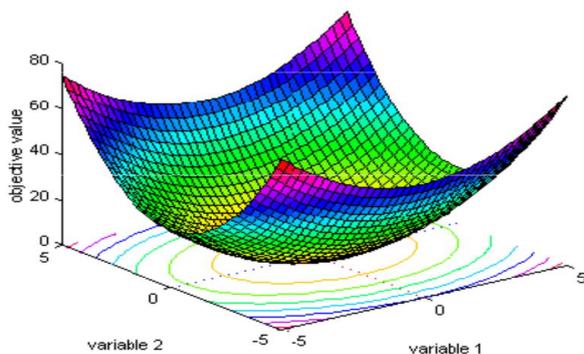


Fig. 3 Plot for the axis parallel hyper ellipsoid

In the study, the population size for the algorithm was assumed 50 and 100. The dimensionality of the function is selected as 10, 20, and 30. From Fig. 5 and Fig. 5 it was inferred that the performance of MEAM was significantly better in comparison to its counterpart algorithm that is EAM, SADE and PSO-TVAC. The fitness value obtained by PSO-TVAC was less in comparison to that of the SADE algorithm.

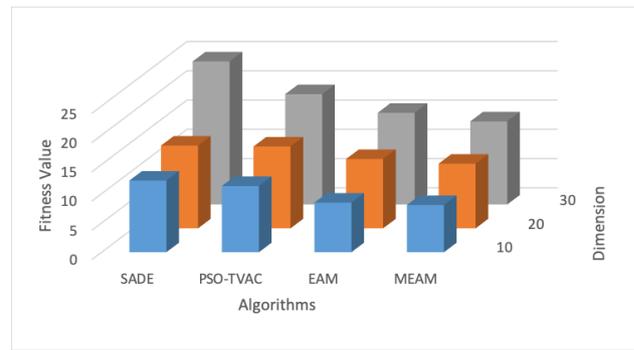


Fig. 4 Comparison of MEAM with SADE, PSO-TVAC and EAM for minimizing Axis Parallel hyper ellipsoid function over the population size of 50

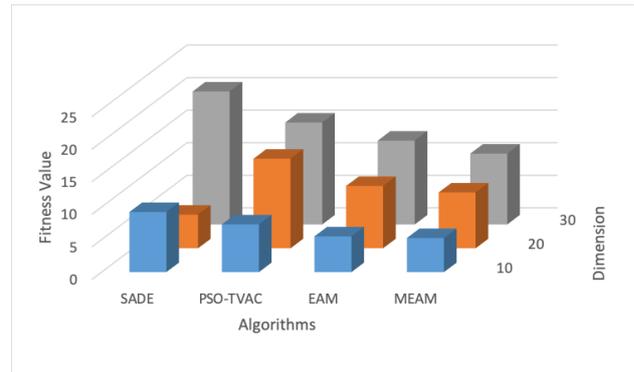


Fig. 5 Comparison of MEAM with SADE, PSO-TVAC and EAM for minimizing axis Parallel hyper ellipsoid function over the population size of 100

B. Sphere Function

Sphere function has been widely used by I. Rechenberg in the evolutionary strategies. Sphere function is a simple unimodal function and is separable. The sphere function is intensive convex function. Fig. 6. highlights the curve of the sphere function. To study the performance of randomized algorithms, over sphere function, the population size was selected as 50 and 100 respectively. The dimensionality of the space function was varied from 10 to 30 with the interval of 10. Result obtained for the sphere function has varied with the change of dimension, hence no algorithm can be stated better than the other. Fig. 7 and Fig. 8. shows the result for the fitness function on population size of 50 and 100 respectively.

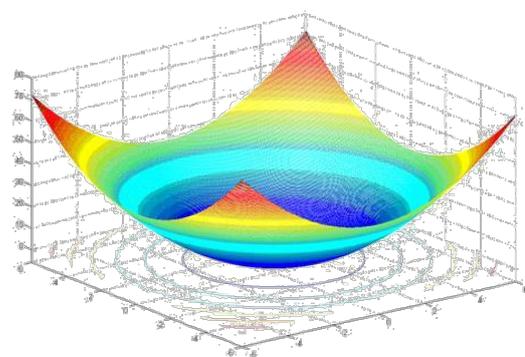


Fig. 6 Plot for sphere function

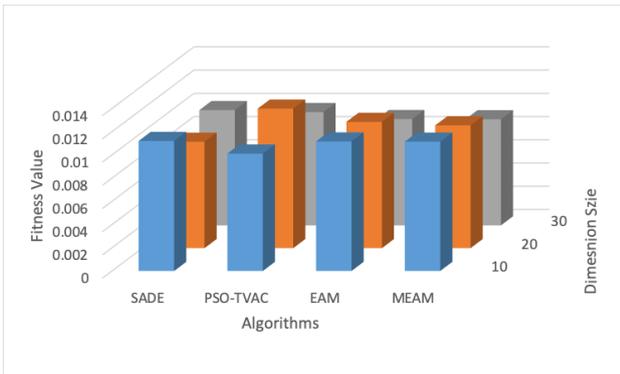


Fig. 7 Comparison of MEAM with SADE, PSO-TVAC and EAM for minimizing here function over the population size 50

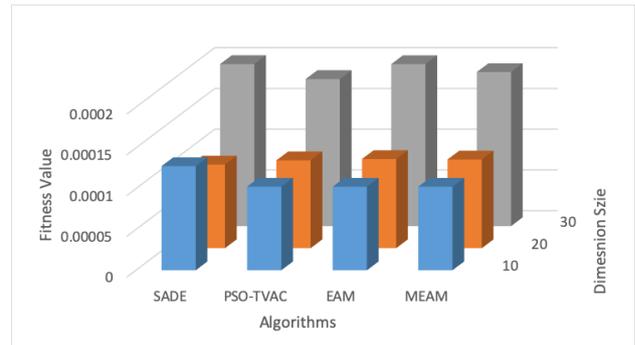


Fig. 10 Comparison of EAM with SADE, PSO-TVAC for minimizing Quartic function over the population size 50

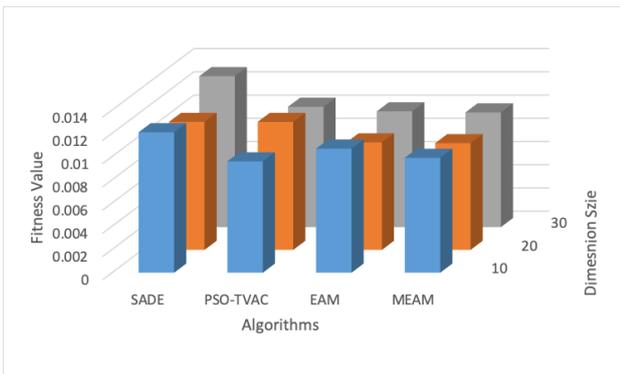


Fig. 8 Comparison of MEAM with SADE, PSO-TVAC and EAM for minimizing Sphere function over the population size 100

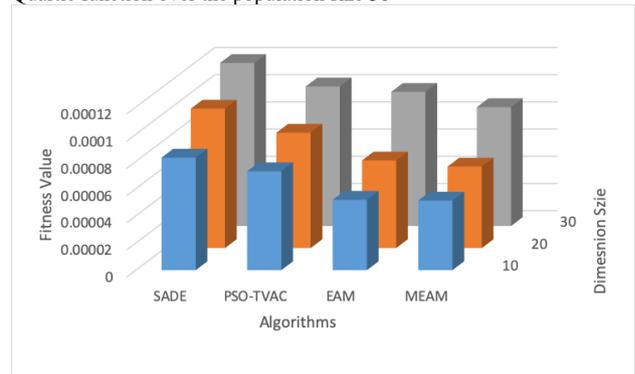


Fig. 11 Comparison of EAM with SADE, PSO-TVAC for minimizing Quartic function over the population size 100

C Quartic Function

Quartic function is a unimodal function of degree four. The quartic function is considered important as in most of the optimization problems, the fitness functions are generally of degree four.

Fig. 9 describes the representation of the quartic function. Quartic function are having higher complexity in comparison to the sphere function. For the assessment of MEAM and other three randomized algorithms, the population size with 50 and 100 are tested on the dimensionality of 10, 20 and 30. The results obtained at population size 50 were non-comparable and are represented in Fig.11. However, EAM has performed better in comparison to the other two algorithms for the population size of 100. Fig. 11 describes the fitness value at different dimensionality for three different algorithms.

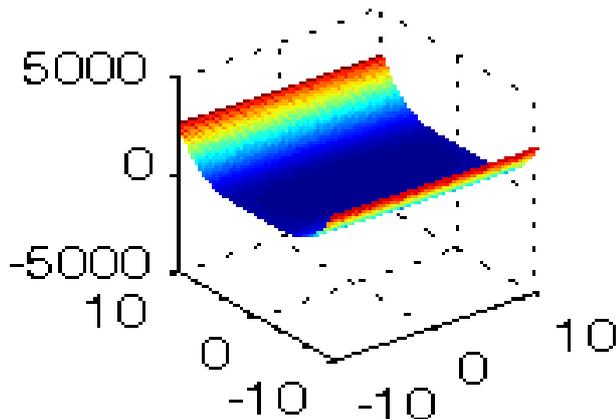


Fig. 9 Plot describing the Quartic function

3.3 Multimodal Function

While designing the optimization algorithm for the multimodal function, the algorithm should have more tendency for exploration in the starting iterations. The reason for extra explorative is to identify the multiple regions for exploration. MEAM has used the same technique and has proved to be beneficial for the multimodal algorithms. The four multimodal functions have been used for the evaluation of MEAM with the three other algorithms. MEAM has better result in comparison to the other algorithms on Rastrigin function, Griewenk function has also converge at the two dimension that are D=10 and D=20. However, the MEAM algorithm could not converge for Schwefel and Zarkharov function for 20 dimensions the same is the case of other two compared algorithms.

3.3.1 Performance of EAM on multimodal benchmark functions:

EAM has been executed and compared with the other two algorithms on the four benchmark functions discussed below:

A. Rastrigin Function

This function has the capability to operate with N different input variables. The function is described by the equation 3 as given below.

$$f(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i + 10)] \quad (3)$$

Fig. 12 it can be observed that Rastrigin function has many local minima within the search area of (-5.12, +5.12). However, the value of global minima as defined in Table 2 is at x* =0 and the function will evaluate to the value zero at this point.

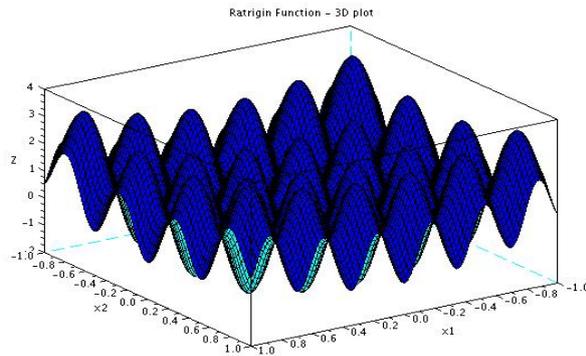


Fig. 12 Curve obtained for Rastrigin

Rastrigin Function is the modified and complex version of De Jong’s function. The complexity of De Jong’s function is increased with summation of cosine function, which in turn adds many local minima. Analogous to unimodal functions the Rastrigin function is evaluated for two different population sizes that are 50 and 100. Each of the three algorithms was compared on both population size on three dimensions that are 10, 20 and 30 respectively. From Fig. 13 and

Fig. 14, it can be observed that MEAM has provided better results for population size 100 for all the three dimensions under consideration. However, at the small population size of 50, it was difficult to say which algorithm is best.

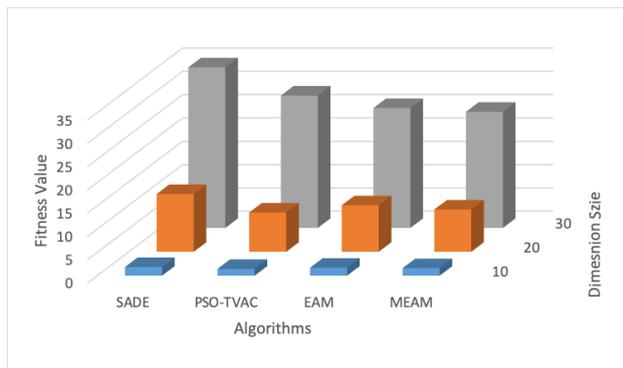


Fig. 13 Comparison of MEAM with SADE, PSO-TVAC, EAM for minimizing Rastrigin over the population size of 50

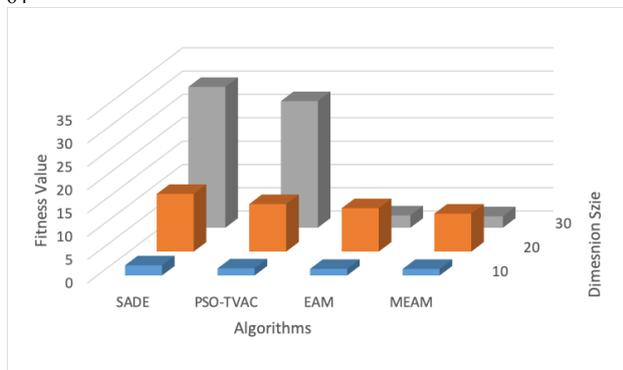


Fig. 14 Comparison of MEAM with SADE, PSO-TVAC and EAM for minimizing Rastrigin over the population size of 100

B. Griewank Function

Like the Rastrigin function, this function can also operate on n input variables. The mathematical representation of Griewank function is described in the equation 4

$$f(x) = \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

(4)

Again from equation 4, it can be observed that there is the component of cosine function which will ultimately introduce local minima to the search space. Fig. 14 describes the plot of the Griewank function with a range of input variable from (-600,+600).

Similar to Rastrigin function the global minima for Griewank Function is found at $x^* = 0$ and the function evaluation results in the value equal to zero. The main difference between the Griewank and Rastrigin function is the spread of the local minima due to the change of cosine function used. The result comparison of MEAM shows that at a lower dimension and small population the results were not good. However, for the remaining dimensions that 20 and 30 and high population size the result obtained was better in comparison to both SADE and PSO-TVAC algorithms. Fig.17 and Fig. 18 narrates the same story.

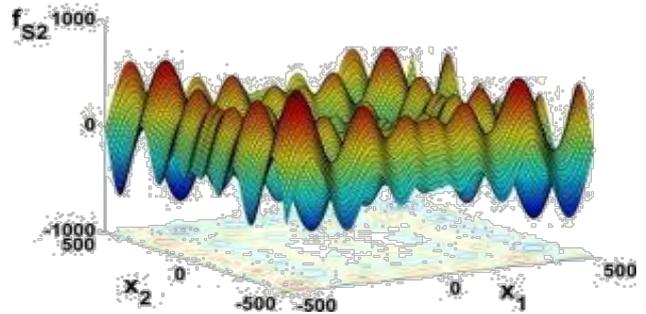


Fig. 15 Graphical representation of Griewank Function

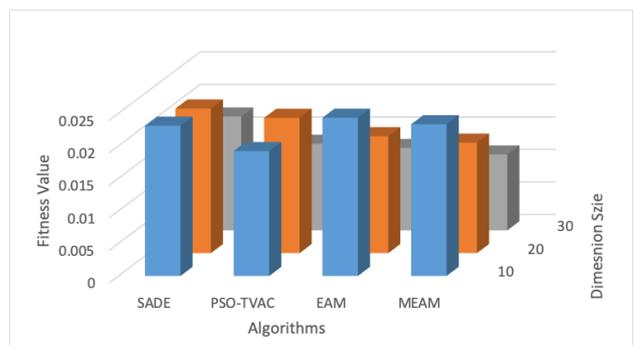


Fig. 15 Comparison of MEAM with SADE, PSO-TVAC and EAM for minimizing Griewank over the population size of 50

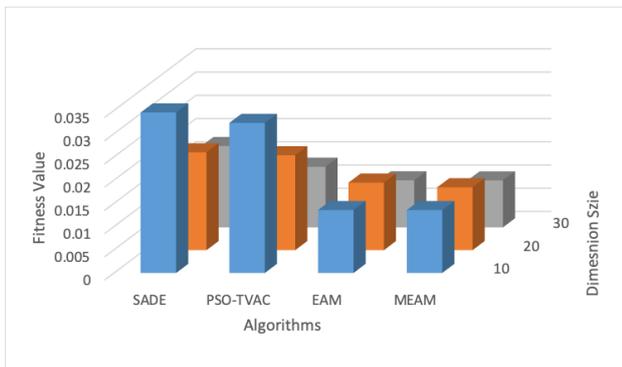


Fig. 16 Comparison of MEAM with SADE, PSO-TVAC and EAM for minimizing Griewank over the population size of 100

C. Schwefl Function

This function has the capability to operate on a large number of variables. The mathematical representation of the Schwefel function is described in equation 5. The function uses the sine domain instead of cosine as discussed in the previous two function. This function is considered as the most important function for benchmarking as this function has the global minima is far from the local minima due to which the function will converge in the wrong direction. Again the MEAM algorithm could not perform well on the 2 dimensions on the population size of 50 but there was significant improvement when the population size was increased to 100.

$$f(x) = \sum_{i=1}^n -xi \sin(\sqrt{|x_i|}) \ln \quad (5)$$

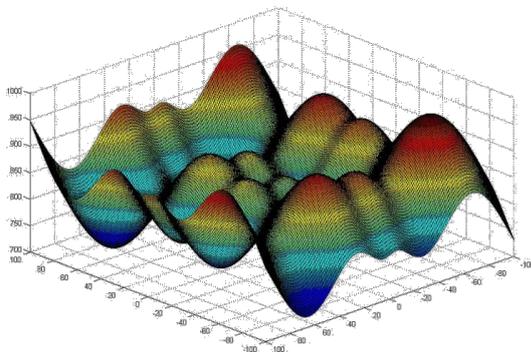


Fig. 17 Graphical representation of Schwefl

Fig. 17 describes the Schwefl function for the range (-500, 500) and global minima could be found at x*= 1 and fitness value is zero. Fig. 18 20 and Fig. 21 describes the fitness value of three algorithms on two dimensions and population size 50 and 100 respectively.

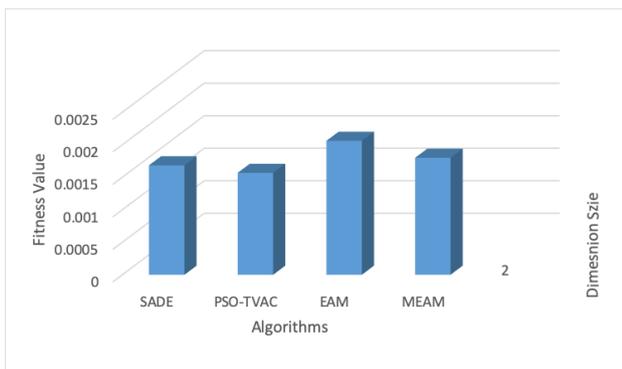


Fig. 18 Comparison of MEAM with SADE, PSO-TVAC and EAM for minimizing Schwefel function over the population size of 50

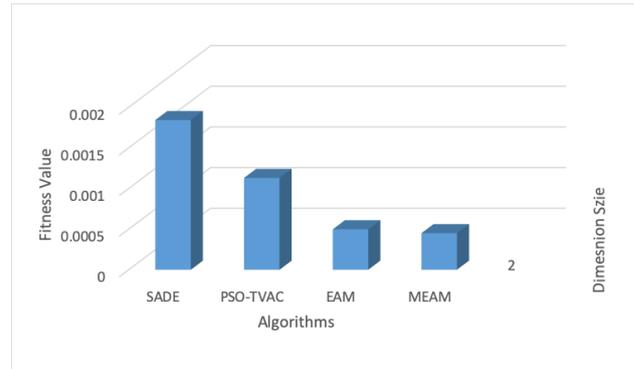


Fig. 19 Comparison of MEAM with SADE, PSO-TVAC and EAM for minimizing Schwefel function over the population size of 100

D. Zakharov Function

Unlike the previous three multimodal functions, the Zakharov function is a kind of polynomial function of degree four, which operates on n variables. This function is described in the equation 6:

$$F(x) = \sum_{i=1}^n x_i^2 + \left(\sum_{i=1}^n 0.5x_i \right)^2 + \left(\sum_{i=1}^n 0.5x_i \right)^4 \quad (6)$$

This function has many local minima in the range of (-5,10) as described in Fig. 20. However, x*=0 is the optimal point and value after the evaluation comes out to be zero.

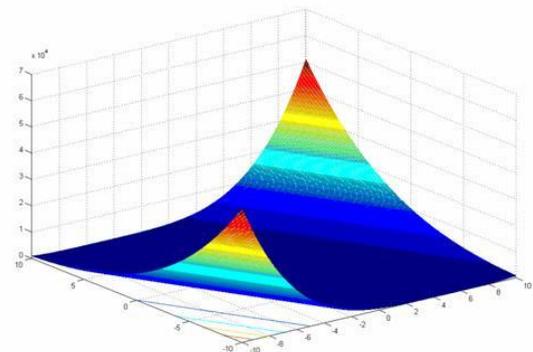


Fig. 20. Graph Describing the Zakharov Function

For assessing the performance of the MEAM, Zakharov is executed with a population size of 100 on three dimensions that are 10, 20 and 30. Fig. 21 narrates that MEAM and EAM are marginally better than the other two algorithms that are part of the study.

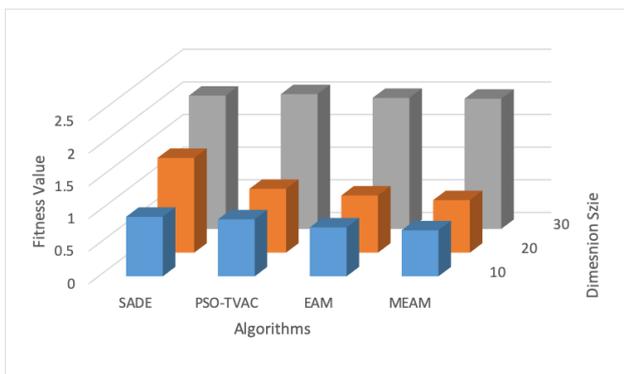


Fig. 21 Comparison of EAM with SADE, PSO-TVAC for minimizing Zakharov function over the population size of 100

4. Real life application of MEAM

In order to test the performance of the algorithm, we have applied the MEAM to generate the test cases in the regression testing. Test case minimization have been done by numerous techniques like linear programming, decision graphs, delta debugging and using the heuristic techniques [16, 17]. This type of testing is done to identify the faults in the modified version of the code termed as the delta version of the code. As the actual source code is already available and has been toughly tested, so the objective of the problem is to generate the test cases that actually exploit the area affected by the modifications performed in the code. As in the past various meta-heuristic approaches have been used for the test case generations and an attempt for the regression testing is also performed in PSO-TVAC [18]. This opens the area of application of heuristic search algorithms in the regression test case generations for the software products.

According to the MEAM algorithm, initially, the random population is to be generated with the size defined as the input to the algorithm. In the current example, the population size is assumed as five. Assuming the population to be represented as described in the set 'P'. Let this population is represented by the integer values 16 bits. These decoded values are used to generate the test suite that can be assigned fitness using the gcov tool based on the code coverage of the code.

The base program(P) which has already being tested on the tested on the test suite (T). The delta code is the modified code of the same program (P). The knowledge about the base code and delta code is utilized by the algorithm to target the area where the modification is done in the program. The heuristic algorithm will generate the temporary test suites, which are being evaluated using the test suite evaluator, and after the specified target of heuristic algorithm is achieved, it will return the best-designed test suite to be used for the evaluation of the system. To study the performance of the approach as we require a tool that helps us to get the evaluation of the test suites. The code coverage can be used as one of the evaluators for the evaluation of the test case. We have used a tool named gcov[19] for this analysis.

The C program accessed from Software-artifact Infrastructure Repository (SIR) [203] is used to develop the

new test cases for the delta version using the fitness function as statement coverage returned from gcov file.

From the comparison of results it can be observed that the performance of MEAM is better or comparable to the other approaches.

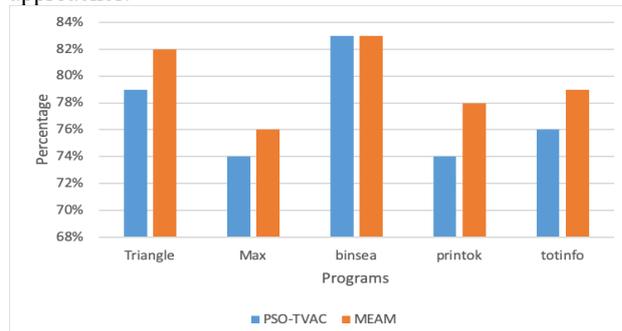


Fig. 24 Comparison of MEAM and PSO-TVAC on different Programs for code coverage

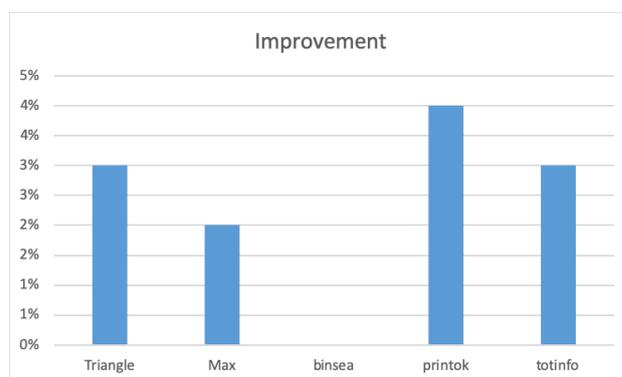


Fig. 25 Describes the comparison of improvement between Pso-TVAC and MEAM

5. Conclusion

The performance of MEAM is better than its counterparts that are in comparison that is a variant of the DE algorithm (SADE) and the variant of PSO (PSO-TVAC) algorithm. However, in some cases the performance is comparable. From the results, it is proved that the auto-adjusting property of MEAM has boosted the exploration power of the algorithm. From the evaluation of the algorithm on the benchmark of different population sizes and different dimensions, it was observed that MEAM has the minimum standard deviation. This verifies that MEAM is robust and stable in comparison to other existing algorithms under comparison. From the real application of MEAM on the problem of test case generation it has been observed that the MEAM has provided the results with the improvement of 4% even though there are the case like binsea program where the performance of the MEAM is to be studied and can be further enhanced in the future.

This is an Open Access article distributed under the terms of the Creative Commons Attribution License



References

- [1] O. K. Erol and I. Eksin, "A new optimization method: big bang--big crunch," *Adv. Eng. Softw.*, vol. 37, no. 2, pp. 106–111, 2006.
- [2] F. Hoffmeister and T. Bäck, "Genetic algorithms and evolution strategies: Similarities and differences," in *International Conference on Parallel Problem Solving from Nature*, 1990, pp. 455–469.
- [3] M. S. Arumugam, M. V. C. Rao, and A. W. C. Tan, "A novel and effective particle swarm optimization like algorithm with extrapolation technique," *Appl. Soft Comput.*, vol. 9, no. 1, pp. 308–320, 2009.
- [4] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm

- optimization,” *Swarm Intell.*, vol. 1, no. 1, pp. 33–57, 2007.
- [5] D. E. Goldberg, “Genetic algorithms in search, optimization, and machine learning, 1989,” *Read. Addison-Wesley*, 1989.
- [6] A. K. Qin, V. L. Huang, and P. N. Suganthan, “Differential evolution algorithm with strategy adaptation for global numerical optimization,” *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 398–417, 2009.
- [7] E. Mezura-Montes, J. Velázquez-Reyes, and C. A. Coello Coello, “A comparative study of differential evolution variants for global optimization,” in *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, 2006, pp. 485–492.
- [8] Y. Shi and R. Eberhart, “A modified particle swarm optimizer,” in *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*, 1998, pp. 69–73.
- [9] H. Fan and Y. Shi, “Study on Vmax of particle swarm optimization,” in *Proc. Workshop on Particle Swarm Optimization, Purdue School of Engineering and Technology*, 2001.
- [10] D. Karaboga and B. Basturk, “A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm,” *J. Glob. Optim.*, vol. 39, no. 3, pp. 459–471, 2007.
- [11] B. Basturk and D. Karaboga, “An artificial bee colony (ABC) algorithm for numeric function optimization, IEEE Swarm Intelligence Symposium 2006, May 12--14, 2006,” Indianapolis, Indiana, USA, 2006.
- [12] M. Giesbrecht and C. P. Bottura, “Recursive Immuno-Inspired Algorithm for Time Variant Discrete Multivariable Dynamic System State Space Identification,” *Int. J. Nat. Comput. Res.*, vol. 5, no. 2, pp. 69–100, 2015.
- [13] G. Klepac, “Particle Swarm Optimization Algorithm as a Tool for Profile Optimization,” *Int. J. Nat. Comput. Res.*, vol. 5, no. 4, pp. 1–23, 2015.
- [14] H. Tang, L. Xie, and S. Xue, “Usage of Comprehensive Learning Particle Swarm Optimization for Parameter Identification of Structural System,” *Int. J. Nat. Comput. Res.*, vol. 5, no. 2, pp. 1–15, 2015.
- [15] D. P. Kanungo, J. Nayak, B. Naik, and H. S. Behera, “Hybrid clustering using elitist teaching learning-based optimization: An improved hybrid approach of TLBO,” *Int. J. Rough Sets Data Anal.*, vol. 3, no. 1, pp. 1–19, 2016.
- [16] W. E. Lewis, *Software testing and continuous quality improvement*. Auerbach publications, 2017.
- [17] G. K. Saha, “Understanding software testing concepts,” *Ubiquity*, vol. 2008, no. February, p. 2, 2008.
- [18] S. Tiwari, K. K. Mishra, and A. K. Misra, “Test Case Generation for Modified Code Using a Variant of Particle Swarm Optimization (PSO) Algorithm,” in *2013 10th International Conference on Information Technology: New Generations*, 2013, pp. 363–368.
- [19] “gcov tool.” [Online]. Available: <http://gcc.gnu.org/onlinedocs/gcc/Gcov.html>.
- [20] Holland, John Henry. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.
- [21] Hoffmeister, Frank, and Thomas Bäck. "Genetic algorithms and evolution strategies: Similarities and differences." In *International Conference on Parallel Problem Solving from Nature*, pp. 455-469. Springer, Berlin, Heidelberg, 1990.
- [22] Civicioglu, Pinar, Erkan Besdok, Mehmet Akif Gunen, and Umit Haluk Atasever. "Weighted differential evolution algorithm for numerical function optimization: a comparative study with cuckoo search, artificial bee colony, adaptive differential evolution, and backtracking search optimization algorithms." *Neural Computing and Applications* (2018): 1-15.
- [23] Chatterjee, Amitava, and Patrick Siarry. "Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization." *Computers & operations research* 33, no. 3 (2006): 859-871.
- [24] Wund, Matthew A. "Assessing the impacts of phenotypic plasticity on evolution." (2012): 5-15.