# An Experimental Analysis of Meta Heuristic Techniques on Unimodal and Multimodal Functions

**Anju Bala, Priti**

***Abstract*: The advancement in the technology leads to the increase in the complexity of the problems. The traditional heuristic algorithms are not suitable for the optimized results of such complex problems. This leads to generation of Meta heuristic techniques which incorporate the exploration as well as the exploitation search. This paper studies different state of art Meta heuristic techniques like ant colony optimization, particle swarm optimization, differential evolution and genetic algorithm. This paper also covers different stable modified version of these techniques and implements the same to analyze the performance on different unimodal and the multimodal functions. The analysis clearly signifies the use of Meta heuristic techniques based on application.**

***Index Terms*: *Exploration, Exploitation, Meta-heuristic, Multimodal and Unimodal***

## I. INTRODUCTION

The field of optimization has become a booming research area during last few years as countless problems have been formulated as an optimization problem. The optimization problems are complex in nature that's why traditional tools and techniques used for solving them didn't result into efficient solution. The need for solving real time problems within short span of time, has attracted the researcher towards the efficient, fast and accurate heuristic algorithm. Meta heuristics, a general level search strategy used to guide or modify the heuristic by exploring and exploiting the search space for complex optimization problems. During the journey of optimal solution, Meta heuristic algorithm passes through two phases: diversification and intensification. Diversification phase explores the search space in all direction to achieve the global solution, whereas Intensification phase focuses on achieving good solution within local specific region.[1]

In the last few years, a lot of Meta heuristic algorithms have been suggested and classified as derivation based and derivation free, trajectory based and population based, deterministic and stochastic, hybrid Meta heuristic, multi objective, nature inspired, cooperative meta heuristic, parallel meta heuristic etc.[2]. Metaheuristic algorithms are found to be fruitful for solving real time complex optimization problem due to their ease of implementation, avoidance of local optima, ability to solve wide variety of problems. Meta heuristic algorithms solves a wide range of optimization problems from design to implementation and every field of daily life. Most of the optimization problem are associated with lot of constraints where Meta heuristics algorithms shows efficient result towards obtaining near optimal solution. Depending upon the complexity, constraints, span of time required for solving the real world problem Meta heuristics algorithms are broadly classified into single objective and multi objective. Single objective Meta heuristic are single solution based where a single solution is improved in successive iterations in different domain, whereas multi objective Meta heuristic is more complex and having more than one objective where the optimal solution is not a single value but a pool of values, each of which has self-optimal decision. [3]. Single solution based metaheuristic includes Simulated Annealing (SA), Local search etc. Multi objective meta heuristic techniques are also termed as population based where the population characteristic guide the search space to achieve number of candidate solution including Genetic Algorithms (GA), Differential Evolution (DE) multi objective (NSGA II, modified DE), Ant Colony Optimization (ACO), Particle Swarm optimization (PSO), Artificial Bee colony (ABC), multi objective ACO, modified PSO and modified ABC.

## II. META HEURISTIC ALGORITHM

Meta heuristic algorithms possesses a number of feature which are considered as necessary aids for solving complex and real life optimization problems. Meta heuristics based algorithm solves the problem in heuristic way in combination with multi agent system[4].Swarm intelligence provides a suitable base for Meta heuristic algorithms due to some unique features including collective intelligence, decentralized, self-organized structure, overall decision making .Collective intelligence between homogenous agents is represented by the interaction between them to exchange any kind of local information without centralized control of any of individual agent. Swarm intelligence algorithm are found to be scalable, fault tolerant and parallel for solving optimization problem.

There exists a number of algorithms based on Meta heuristic concept, some of

**Revised Manuscript Received on May 22, 2019.**

**Anju Bala**, Research Scholar, Dept. of Computer Science and Applications, M.D.U, Rohtak, India.

**Priti,** Assistant Professor, Dept. of Computer Science and Applications, M.D.U, Rohtak, India.



*Retrieval Number: H10350688S319/19©BEIESP*

*Published By:*
*Blue Eyes Intelligence Engineering*
*& Sciences Publication*

the most popular algorithm are ACO, PSO, ABC [5] etc.

### A. *Ant Colony Optimization (ACO)*

ACO is staying today as one of the most important nature inspired Meta heuristic algorithm which solves optimization problem within reasonable amount of time. ACO algorithm have been applied over diverse fields like traffic congestion, feature selection in data mining, transportation, manufacturing, telecommunication network. It is based on foraging behavior of real ants.

ACO procedure starts with a pheromone model which is defined as a set of pheromone values represented by Γ. A set of solution components € is defined to form the final optimal solution. Initially each solution starts with an empty sequence s= (). Candidate solution is constructed using pheromone values from the pheromone model, it is done by selecting the components one after the other to obtain the final solution, assembling them with pheromone model. The selection of solution component from set € is performed probabilistically by using the (1).

$$P(€_i \mid s) = \frac{[\Gamma_i]^{\alpha}.[\theta(€_i)]^{\beta}}{\sum_{€_j \,\varepsilon\, €}[\Gamma_j]^{\alpha}.[\theta(€_j)]^{\beta}} \; €_i \,\varepsilon\, € \quad (1)$$

Pheromone model is updated with the help of candidate solution to achieve high quality solution. Repeat the process till the final optimal solution is obtained without violating any of the problem constraints.[6][7].

### B. *Modified ACO (MACO)*

ACO is showing its applicability over a wide variety of problems

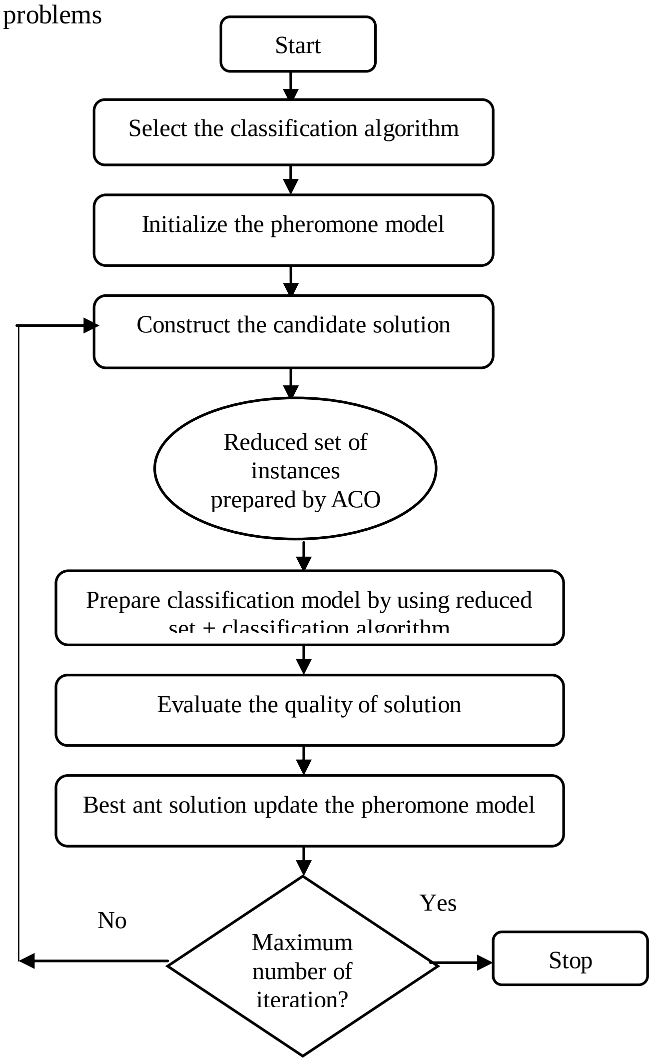


Figure 1: Flowchart of MACO

Including choosing efficient novel routes for vehicles, scheduling of activities, cluster analysis, classification, instance selection, discriminant analysis (subclass of classification), traffic signal timing optimization, timetable management etc. [8]. The basic work flow of ACO is also modified in some way to make it suitable for different application which is as described in figure 1. ACO for instance selection produce the most effective classification model for predictive accuracy. In MACO, ACO is used in the initial phase of classification for evaluation of candidate solution then final classification model is built by using the reduced set produced by ACO [9] .

### C. *Particle Swarm Optimization (PSO)*

For solving complex optimization problem PSO is found to be one of the popular Meta heuristic algorithm. It is applicable over a variety of problems including data mining, telecommunication, optimization, signal processing, pattern recognition, robotics, artificial neural networks, games, automatic target detection etc. [10]. PSO is showing its applicability over any kind of problem constrained or unconstrained, single objective or multi-objective optimization problem having multiple solution. Emulation in PSO is based on swarm such as bird flocking, where position of particles is solution itself which is modified in each successive iteration depending upon the communication between the particles to optimize the solution[11].The process initiates the population of particle (N) with some random values for position p and velocity v. Then optimization function $F(opt)$ is decided whether needs to be minimized or maximized. Two best values for the particle $p(best)$ and $g(best)$ are calculated for the particle, where $p(best)$ its own best solution is till now in the history (fitness) and $g(best)$ is best value obtained by any particle in population till now. Comparison of fitness values is done with $p(best)$ value, if the current fitness value is found to be better than $p(best)$ value in history then current value is chosen as new $p(best)$ for calculating new $g(best)$ value, particle with best fitness value among the whole population is selected. Update the velocity and position of particle is updating according (2) & (3).

$v_n = v_n + k_1$* rand ()*($p(best)$-$x_n$) +$k_2$* rand ()*($g(best) - x_n$) (2)

$x_n = x_n + v_n$ (3)

Where $v_n$ is the velocity and $x_n$ is the position of particle after n iterations. $k_1$ And $k_2$ are two constants whose value is set as 2 an ideal value to pull the particle towards its best position. Rand () is random number between (0, 1). Repeat the process, till the desired optimal solution is not obtained.

In this way a particle proceed towards the





Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

optimized solution after updating the candidate solution in the successive iteration and stops when either the pre decided number of iterations is met or the solution is not further modified in successive iterations.

### *D. Modified PSO (MPSO)*

The pitfalls of PSO algorithm including partial optimism, getting trapped in local optima, premature convergence, etc. can be cured up to a great extent by using variants of PSO by changing different parameters included in this algorithm, hybridizing PSO with other available algorithm and expansion of search space. Basic variants of PSO can be obtained by incorporating the changes in velocity to control the global expansion, in momentum to optimize the convergence, constriction coefficient to stable the convergence [7]. The complete process of modified PSO is as depicted in the figure 2:

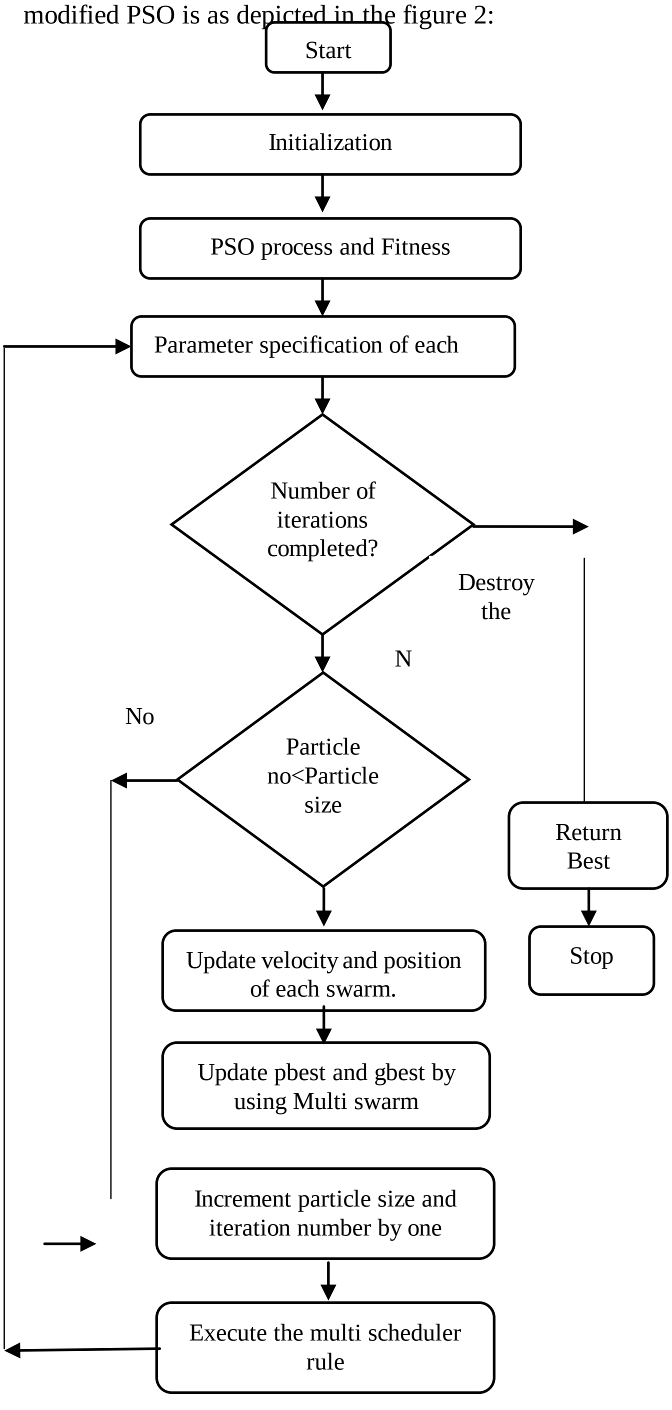


Figure 2: Flowchart of

For removing premature convergence, a swarm of particles is sampled randomly which is further divided into sub swarms which are evolved on PSO. Any time reshuffling of particles to sub swarms for information sharing results into Improved particle swarm optimization algorithm which proves to be more efficient in terms of exploration and exploitation ability and to achieve global optimum [12]. MSPSO (Multi Swarm Particle Swarm Optimization) algorithm is used to describe the competition among the swarm by using multi swarm scheduler for gathering the results from multiple swarms which was previously ignored in PSO. Survival of fittest is used to decide the destruction or reservation of swarms[13].

Improved PSO by using different variation in inertia weight and constriction factor found to be beneficial for its applicability in text feature selection in data mining. The equation for velocity and position of particle are modified in variants of PSO to satisfy the requirements[14].

$v_n = 0.9 * v_n + 2 *$ rand ()*$(p(best) - x_n)$ + 2*rand ()*$(g(best) - x_n)$ (4)

$x_n = x_n + v_n$ (5)

Where the inertia weight w=0.9 and constant are having fixed value of 2.

Concisely, modified PSO is applicable in number of fields to solve the optimization problem based on PSO.

### *E. Genetic Algorithm (GA)*

To solve real world optimization problems of high complexity, GA, one of the heuristic optimization search technique inspired by concept of evolution in nature is used. Due to its modular nature, ease of implementation, reusability of standard components to design the solution it has become popular among researchers [15].A population of individual is maintained in the search space to represent possible solution for a problem, linked with chromosomes (solution) composed of genes (variable). Optimal solution evolves in iterative manner by combining the best existing solution depending upon their fitness value. In each successive iteration newly obtained partial optimal solution replaces the previous one, this iterative production continues till no more efficient solution than the previous one is obtained and then the problem is said to converge with this set of solution. The evolution in GA is based upon three basic operators including (i) Selection (ii) Cross over (iii) Mutation. Selection depends upon the fitness value, better individuals are selected from the whole population. Cross over operator makes GA algorithm as unique algorithm among the other optimization technique. For obtaining better individual good individuals are combined by using this operator. Mutation provides diversity in the population. Along the inherited features, new individual also possess some of their own traits to





*Published By:*
*Blue Eyes Intelligence Engineering*
*& Sciences Publication*

maintain diversity. Genetic algorithm for solving any of the real world optimization works depending upon the problem, generate an initial random population P of chromosomes.

Then, calculate the fitness value, F(c), of each chromosome c in the source population for the selection of better individual. An empty successor population is created and then repeats the following process which includes selection, crossover, mutation and acceptance until P chromosomes have been created. Selection: Using selection operator, select two parent chromosomes, suppose c1 and c2, from the source population based on their fitness value. Better fitness value, more chances to be parent. Crossover: Apply one-point crossover to c1 and c2 to obtain a child chromosome c. Mutation: Apply uniform mutation to c to maintain diversity in the population. Acceptance: Accept this mutated c to new population. Then replace the source population with the successor population. Repeat the process until stopping criteria have not been met [16]. In this way, Genetic algorithm works to generate solution for real world highly complex optimization problem. Genetic algorithm is widely applicable over a range of problems including shortest path routing problem, turbine control system, and channel assignment problem etc.

### *F. Non Dominated Sorted Genetic Algorithm (NSGA)*

In order to solve complex, real world multi objective optimization problem, traditional technique of converting multi objective optimization problem into single objective optimization didn't give efficient outcome as only one particular Pareto solution is discovered in each run. To maintain diversity in solution evolutionary algorithm is used. Multiple Pareto solutions within a single run are possible by using Evolutionary algorithm. NSGA (Non dominated sorted genetic algorithm) is one of the GA which provides a set of Pareto solutions for multi-objective problem simultaneously[17]. . The whole process of NSGA is as depicted in the figure 3.
This algorithm works on same operators such as mutation, cross over and selection but the way of selection makes it different from other GA's. Before the selection is performed, population is ranked according to their non-dominance. Population is divided into sub population of good points. Any number of objectives is solved by using this strategy. This algorithm works on same operators such as mutation, cross over and selection but the way of selection makes it different from other GA's. Before the selection is performed, population is ranked according to their non-dominance. Population is divided into sub population of good points. Any number of objectives is solved by using this strategy
But NSGA suffers from certain pitfalls including. Due to the existing pitfalls of NSGA algorithm, NSGA II, multi objective GA was proposed with the properties of fast non-dominated sorting approach, no need of sharing parameter (parameter less approach), an elitist strategy, efficient constraint handling process, reducing complexity up to great extent, fast crowded distance estimation procedure and simple crowded comparison operator.[18]

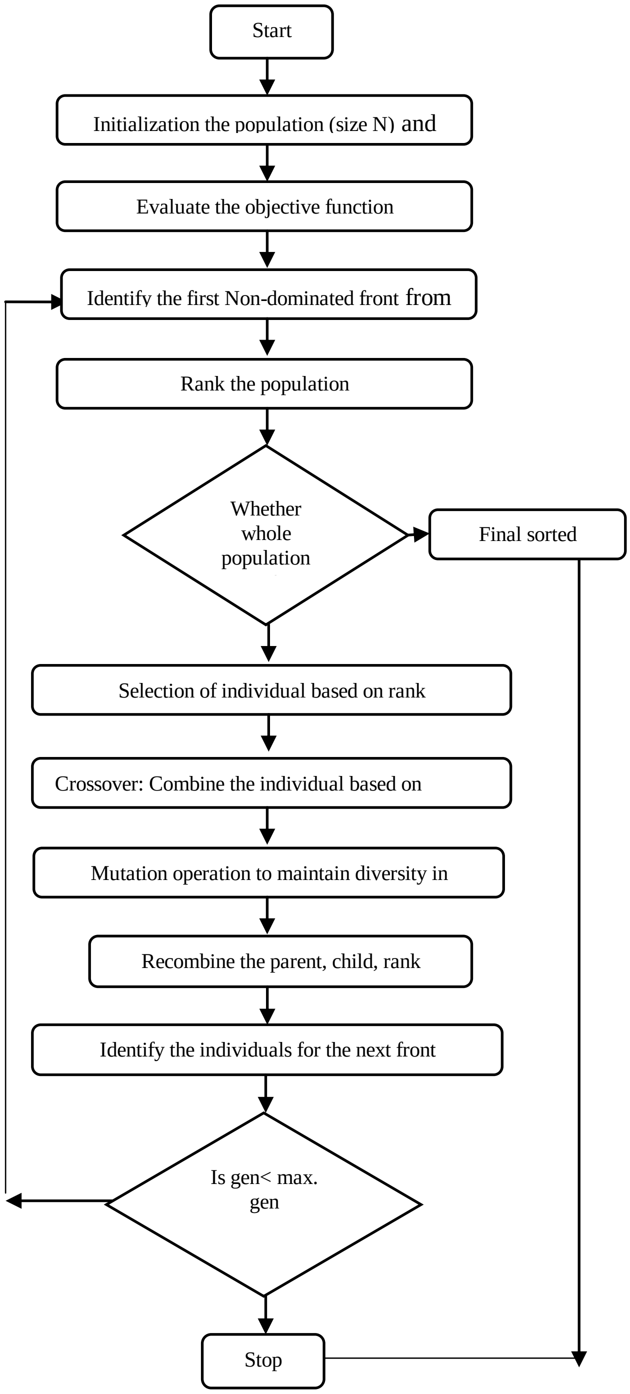


Figure 3: Flowchart for





*Published By:*
*Blue Eyes Intelligence Engineering*
*& Sciences Publication*

### *G. Differential Evolution (DE)*

DE is an evolutionary technique which is proven to be better than others due to a number of features like simplicity, equally applicable in case of integer and discrete optimization problem, fast convergence, global optima regardless of initial parameters.[19]. It is one of the population based algorithm which is effectively used for large scale, flat, multi-dimensional, multimodal and constrained problems. In this algorithm, fitness function doesn't play any significant role as all solutions has equal chances of being selected as parent. It is based on three operators' selection, mutation and cross over like other genetic algorithms. But it relies on mutation operation for performing in a different way than other algorithms. The general framework of the algorithm is

as follows: To optimize a function with a set of real parameters R, Gas generation number and N as population size. The parameter vector takes the following form.

$x_{i,G}$= [ $x_{1,i,G}$, $x_{2,i,G}$, ……. $x_{R,i,G}$] where i =1, 2 …N

Step 1. **Initialization**: In this step, population is generated randomly between lower and upper bounds and initial parameter values are selected.

Step 2. **Mutation**: This step expands the search space. Three vectors $x_{r1,G}$ , $x_{r2,G}$, $x_{r3,G}$ are selected for a parameter vector $x_{i,G}$. Donor vector (DV) is calculated by adding the weighted difference of two vectors to the third one.

$$DV_{i,G+1} = x_{r1,G} + M_f\left(x_{r2,G} - x_{r3,G}\right) \quad (6)$$

Where $M_f$ is a mutation factor with values [0,2].

Step 3. **Recombination**: To obtain successful solution from the previous generation recombination is required. Trial vector (TV) is developed here by using donor vector with the probability P and target vector $x_{i,G}$.

$$TV_{i,j,G+1} = \begin{cases} DV_{j,i,G+1} \text{ if } rand_{j,i} \leq \text{ P or j} = I_{rand} \\ x_{j,i,G} \text{if } rand_{j,i} > \text{ P or j} \neq I_{rand} \end{cases} \quad (7)$$

Where i = 1, 2, 3………N and j= 1, 2…..R

$rand_{j,i}$ approx. $U[0,1]$

Step 4. **Selection**: In this step if the new individual is found to be better than the previous one then it is selected as an offspring for the next generation otherwise it is discarded. This comparison is done on the basis of fitness function as follows:

$$x_{i,G+1} = \begin{cases} TV_{i,G+1} \text{ if f}\left(TV_{i,G+1}\right) \leq \text{f}\left(x_{i,G}\right) \\ x_{i,G} \ \ otherwise \end{cases} \quad (8)$$

Step 5. Repeat step 2, 3, 4 till stopping criteria is met.

In this way the differential evolution algorithm works to solve any optimization problem. A number of variants are available for the mutation operator depending upon those a lot of evolution strategy are constituted including DE/rand/1, DE/best/1, DE/rand/2 etc.[20].

### *H. Modified DE (MDE)*

Depending upon the variants of mutation operator used, DE algorithm is modified in different ways to make it applicable over varieties of problems existing in different domains. To select relevant features from a high dimensional data set, is nothing but feature subset selection. For scoring different evolved features, feature subset along with an optimization algorithm is used. To avoid computational complexity, over fitting problem, Differential Evolution algorithm is used for selecting optimal features from huge data with high accuracy. DE/rand/wt-2/expr mutation scheme is used for feature subset selection, it is found to be one of the best optimizer evolutionary algorithm. A wide range of criteria is used for the evaluation of feature subset such as mutual information, correlation, distance measure etc. Out of which mutual information is of primary importance. Different approaches with greedy search have been suggested, but no one is widely accepted due to their lot of shortcomings. With global search capacity, modified differential algorithm is used with filter approach having mutual information for feature selection. As DE has only few control parameters to change and less space complexity it is widely acceptable and popular. MDE shows better performance than the original one.DE/rand/wt-2/expr mutation scheme is used for feature subset selection, it is found to be one of the best optimizer evolutionary algorithm and works in the following way.[21]

Step 1. Initially wide range of value must be converted into real values between 0 and 1.

Step 2. **Initialization:** Generate the population vector

$x_{i,G}$= [ $x_{1,i,G}$ , $x_{2,i,G}$ , ……. $x_{R,i,G}$] ; i =1, 2 …N

Where G is the generation number, N is the population size, R is set of real parameters.

Step 3. **Mutation**: Calculate the mutant vector/ donor vector by using the following formula:

$$M_{i,G} = \text{F.}\left(x_{i,r1,G} - x_{i,r2,G}\right) + \text{F.}\left(x_{i,r3,G} - x_{i,r4,G}\right)$$

Where $x_{i,G}$ is the target vector for which mutant vector $M_{i,G}$ is generated by adding the weighted difference of four selected vectors.

Step 4. **Recombination/ Cross Over:** Trial vector is generated by using cross over operation between target vector and its mutant vector.

$$TV_{i,j,G+1} = \begin{cases} M_{j,i,G} \text{ if } rand_{j,i} \leq \text{ P or j} = I_{rand} \\ x_{j,i,G} \text{if } rand_{j,i} > P \ or \ j \neq I_{rand} \end{cases}$$

Where P is the probability

Step 5. **Selection:** Comparison between trial and target vector depending upon the fitness function is carried out, first individual vector for the next generation is selected in this step.

Step 6. Repeat the 3, 4, and 5 till the next generation vectors are unchanged.

Step 7. Stop

In this way, modified differential evolution algorithm gives

optimal subset of features without losing the accuracy and performance with the increasing dimension of data over the time. The whole process is as shown in figure 4.



Retrieval Number: H10350688S319/19©BEIESP

*Published By:*
*Blue Eyes Intelligence Engineering & Sciences Publication*

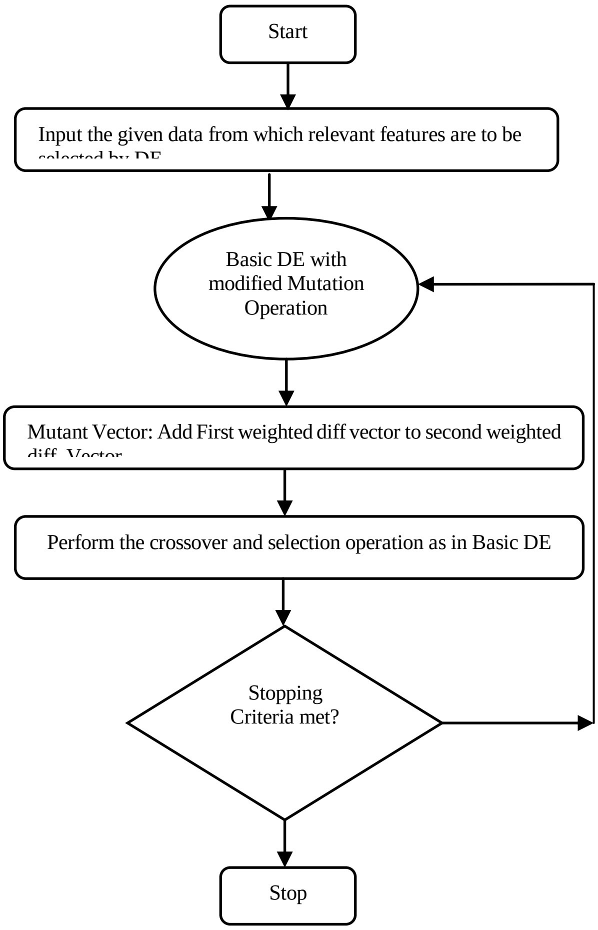


Figure 4: Flowchart of MDE for feature selection

The algorithms discussed in previous section have been implemented using MATLAB and analyzed over different unimodal and multimodal functions described in table 1 and table 2 respectively.

**TABLE 1 : UNIMODAL BENCHMARK FUNCTIONS**

| Acronym | Function | Dim | Range | Value |
|---|---|---|---|---|
| F1 | $f(y)=\sum_{j=1}^{n} jy_j^2$ | 100 | [-100, 100] | 0 |
| F2 | $f(y)=\sum_{j=1}^{n} y_j^2$ | 100 | [-100, 100] | 0 |
| F3 | $f(y)=\sum_{j=1}^{n} jy_j^4+random[0,1)$ | 100 | [-100, 100] | 0 |
| F4 | $f(y)=100(y_1^2-y_2)^2+(y_1-$ $+(y_3-1)^2+90(y_3^2-y_4)^2$ $+10.1((y_2-1)^2+(y_4-1)^2)$ $+19.8(y_2-1)(y_4-1)$ | 10 | [-100, 100] | 0 |
| F5 | $f(y)=\sum_{j=1}^{n-1}[100(y_{j+1}-y_j^2)^2+(y_j-$ | 30 | [-100, 100] | 0 |
| F6 | $f(y)=\sum_{(j=1)}^{(n/k)}(y_{(4j-3)}+10y_{(4j-2)})^2$ $+5(y_{(4j-1)}+y_4j)^2+(y_{(4j-2)}+y_{(4j-1}$ $+10(y_{(4j-3)}+y_4j)^4$ | 24 | [-100, 100] | 0 |
| F7 | $f(x)=-(x_1-1)^2+\sum_{i=0}^{n} i(2x_i^2-x_i$ | 10 | [-100, 100] | 0 |
| F8 | $f(x)=-cos(x_1)cos(x_2)$ $\exp\left(-(x_1-\pi)^2-(x_2-\pi)^2\right)$ | 10 | [-100, 100] | -1 |

**TABLE 2: MULTIMODAL BENCHMARK UNCTIONS**

| Acronym | Function | Dim | Range | Value |
|---|---|---|---|---|
| F9 | $f(x)=\sum_{i=1}^{n} x_i^2 - 10\cos(2\pi x_i)+10$ | 100 | [-100, 100] | 0 |
| F10 | $f(x)=\sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$ | 100 | [-500, 500] | 0 |
| F11 | $f(x)=-\sum_{i=1}^{n} \sin(x_i)\left(\sin\left(ix_i^2/\pi\right)\right)^2$ | 30 | [0, $\pi$] | -4.688 |
| F12 | $f(x)=\left[\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(x_i-a_{ij})^6}\right]^{-1}$ | 30 | [-100, 100] | 0.998 |
| F13 | $f(x)=-\sum_{i=1}^{5}\left[(x-a_i)(x-a_i)^T+c_i\right.$ | 30 | [0, 10] | -10.151 |

Function name along with the range, dimension and the value for each function has been specified in the table 1 and table 2. Each function has been optimized by using NSGA-II, MACO, MDE and the MOPSO on the given range for the specified dimension. The performance has been analyzed using the MATLAB for 500 iterations. The given dimension gives the number of variable (number of ants in MACO, number of particle in MPSO, number of chromosome in NSGA-II) for each algorithm while the range of corresponding function gives lower bound and upper bound. The optimization results for unimodal and multimodal functions have been specified in the table 3 and table 4 respectively.

**TABLE 3: RESULT COMPARISON ON UNIMODAL FUNCTIONS UNITS**

| Functions | NSGA-II | | MACO | |
|---|---|---|---|---|
| | Average | Standard Deviation | Average | Standard Deviation |
| **F1** | 3.24E+00 | 6.58E-01 | 3.51E+03 | 4.88E+03 |
| **F2** | 5.34E-01 | 7.25E-02 | 2.41E+01 | 4.01E+01 |
| **F3** | 4.75E+03 | 1.25E+03 | 1.50E+04 | 7.77E+03 |
| **F4** | 7.40E+00 | 1.74E+00 | 6.39E+01 | 2.00E+01 |
| **F5** | 4.81E+02 | 3.38E+02 | 8.21E+02 | 9.19E+02 |
| **F6** | 3.13E+00 | 8.80E-01 | 1.02E+03 | 3.04E+03 |
| **F7** | 1.45E-01 | 5.85E-02 | 2.95E-01 | 9.19E-02 |
| **Functions** | **MDE** | | **MOPSO** | |
| | **Average** | **Standard Deviation** | **Average** | **Standard Deviation** |
| **F1** | 1.77E-02 | 1.23E-02 | 3.77E+04 | 8.16E+03 |
| **F2** | 3.47E-02 | 8.37E-02 | 6.16E+06 | 2.29E+07 |
| **F3** | 2.14E+04 | 2.14E+04 | 8.63E+04 | 3.41E+04 |
| **F4** | 1.10E+01 | 1.11E+01 | 8.35E+01 | 5.90E+00 |
| **F5** | 7.59E+01 | 7.58E+01 | 1.11E+08 | 3.85E+07 |
| **F6** | 2.12E-02 | 6.86E-02 | 3.24E+04 | 7.01E+03 |
| **F7** | 5.64E-02 | 1.13E-01 | 6.65E+01 | 2.36E+01 |

**TABLE 4: RESULT COMPARISON ON MULTIMODAL FUNCTIONS**

| Functions | NSGA-II | | MACO | |
|---|---|---|---|---|
| | Average | Standard Deviation | Average | Standard Deviation |
| F8 | -1.41E+04 | 5.86E+02 | -7.07E+03 | 7.13E+02 |
| F9 | 6.65E+00 | 2.88E+00 | 1.45E+02 | 2.87E+01 |
| F10 | 5.37E-01 | 1.92E-01 | 1.53E+01 | 3.47E+00 |
| F11 | 1.06E+00 | 1.14E-01 | 3.76E+01 | 6.12E+01 |
| F12 | 8.63E-02 | 9.51E-02 | 1.75E+01 | 8.61E+00 |
| F13 | 1.64E-01 | 1.39E-01 | 2.05E+07 | 9.17E+07 |
| **Functions** | MDE | | MOPSO | |
| | **Average** | **Standard Deviation** | **Average** | **Standard Deviation** |
| F8 | -9.81E+03 | 4.57E+02 | -4.45E+03 | 6.72E+02 |
| F9 | 8.94E+01 | 8.63E+00 | 3.03E+02 | 2.81E+01 |
| F10 | 4.68E-02 | 6.32E-02 | 1.95E+01 | 6.24E-01 |
| F11 | 5.29E-02 | 1.07E-01 | 3.21E+02 | 7.83E+01 |
| F12 | 2.07E-02 | 4.55E-02 | 1.97E+08 | 1.18E+08 |
| F13 | 9.31E-02 | 1.94E-02 | 3.77E+08 | 1.98E+08 |

The evaluation has been done for ten times and the average result of ten evaluations along with the standard deviation has been displayed in the table 3 and table 4 for unimodal and multimodal functions respectively. The results clearly show that results of NSGA-II and MDE are better as compared to the MPSO and the MACO. The NSGA-II shows the better results on few functions as compared to MDE while for the remaining functions MDE shows better results.

## III. CONCLUSION

This paper implements NSGA-II, modified ACO, PSO and modified DE optimization. This paper analyzes the performance of above specified algorithms on different unimodal and the multimodal functions. The average results of ten evaluations along with the standard deviation clearly show that the performance of NSGA-II and MDE is better on these functions as compared to the MACO and the MPSO. While the NSGA and MDE exhibits better performance with respect to each other on different functions. It means the NSGA-II and MDE can be used for different applications depending upon the performance of corresponding optimization function. In future NSGA-II and MDE can be applied to different optimization problem and both can be hybridized to enhance the optimization. The hybrid NSGA-II and MDE may perform better than the NSGA-II as well as MDE.



Retrieval Number: H10350688S319/19©BEIESP

*Published By:*
*Blue Eyes Intelligence Engineering & Sciences Publication*

## REFERENCES


[1] R. Rajakumar, P. Dhavachelvan, and T. Vengattaraman, "A survey on nature inspired meta-heuristic algorithms with its domain specifications," *Proc. Int. Conf. Commun. Electron. Syst. ICCES 2016*, 2016.

[2] M. G. Torres, "Metaheuristics in Data Mining," no. August, 2008.

[3] R. Rao Kurada, K. K. Pavan, and A. D. Rao, "A Preliminary Survey on Optimized Multiobjective Metaheuristic Methods for Data Clustering Using Evolutionary Approaches," *Int. J. Comput. Sci. Inf. Technol.*, vol. 5, no. 5, pp. 57–77, 2013.

[4] X. S. Yang, S. Fong, X. He, S. Deb, and Y. Zhao, "Swarm Intelligence: Today and Tomorrow," *Proc. - 2016 3rd Int. Conf. Soft Comput. Mach. Intell. ISCMI 2016*, pp. 219–223, 2017.

[5] S. C. Chu, H. C. Huang, J. F. Roddick, and J. S. Pan, "Overview of algorithms for swarm intelligence," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6922 LNAI, no. PART 1, pp. 28–41, 2011.

[6] C. Blum, "{A}nt colony optimization: {I}ntroduction and recent trends," *Phys. Life Rev.*, vol. 2, no. 4, pp. 353–373, 2005.

[7] M. Dorigo and C. Blum, "Ant colony optimization theory: A survey," *Theor. Comput. Sci.*, vol. 344, no. 2–3, pp. 243–278, 2005.

[8] H. Ali and A. K. Kar, "Discriminant Analysis using Ant Colony Optimization - An Intra-Algorithm Exploration," *Procedia Comput. Sci.*, vol. 132, pp. 880–889, 2018.

[9] I. M. Anwar, K. M. Salama, and A. M. Abdelbar, "Instance Selection with Ant Colony Optimization," *Procedia - Procedia Comput. Sci.*, vol. 53, pp. 248–256, 2015.

[10] R. C. Eberhart, "AltaVista - Babel Fish Translation," pp. 81–86, 1995.

[11] D. P. Rini and S. M. Shamsuddin, "Particle Swarm Optimization: Technique, System and Challenges," *Int. J. Appl. Inf. Syst.*, vol. 1, no. 1, pp. 33–45, 2011.

[12] C. F. Wang and K. Liu, "An improved particle swarm optimization algorithm based on comparative judgment," *Nat. Comput.*, vol. 17, no. 3, pp. 641–661, 2018.

[13] Y. Liu, G. Wang, H. Chen, H. Dong, X. Zhu, and S. Wang, "An improved particle swarm optimization for feature selection," *J. Bionic Eng.*, vol. 8, no. 2, pp. 191–200, 2011.

[14] Y. Lu, M. Liang, Z. Ye, and L. Cao, "Improved particle swarm optimization algorithm and its application in text feature selection," *Appl. Soft Comput. J.*, vol. 35, pp. 629–636, 2015.

[15] J. McCall, "Genetic algorithms for modelling and optimisation," *J. Comput. Appl. Math.*, vol. 184, no. 1, pp. 205–222, 2005.

[16] R. Malhotra, N. Singh, and Y. Singh, "Genetic Algorithms : Concepts , Design for Optimization of Process Controllers," *Comput. Inf. Sci.*, vol. 4, no. 2, pp. 39–54, 2011.

[17] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, 2002.

[18] Y. Yusoff, M. S. Ngadiman, and A. M. Zain, "Procedia Engineering Overview of NSGA-II for Optimizing Machining Process Parameters," *Rg*, vol. 00, pp. 3978–3983, 2011.

[19] D. Karaboğa and S. Ökdem, "A simple and global optimization algorithm for engineering problems: Differential evolution algorithm," *Turkish J. Electr. Eng. Comput. Sci.*, vol. 12, no. 1, pp. 53–60, 2004.

[20] A. Youyun and C. Hongqin, "Experimental study on differential evolution strategies," *Proc. 2009 WRI Glob. Congr. Intell. Syst. GCIS 2009*, vol. 2, pp. 19–24, 2009.

[21] T. Vivekanandan and N. C. Sriman Narayana Iyengar, "Optimal feature selection using a modified differential evolution algorithm and its effectiveness for prediction of heart disease," *Comput. Biol. Med.*, vol. 90, no. April, pp. 125–136, 2017.


## AUTHORS PROFILE

Author-1 Photo

**First Author** personal profile which contains their education details, their publications, research work, membership, achievements, with photo that will be maximum 200-400 words.

Author-2 Photo

**Second Author** personal profile which contains their education details, their publications, research work, membership, achievements, with photo that will be maximum 200-400 words.

Author-3 Photo

**Third Author** personal profile which contains their education details, their publications, research work, membership, achievements, with photo that will be maximum 200-400 words.



Retrieval Number: H10350688S319/19©BEIESP

*Published By:*
*Blue Eyes Intelligence Engineering*
*& Sciences Publication*