

Memetic Algorithm with SIIO or BFO's Chemotaxis Mechanism as Multi-meme for Function Optimization

D. Jadhav¹ and S. Pattnaik²

¹Dr. Babasaheb Ambedkar Technological University's Institute of Petrochemical Engineering, Lonere, 402103 India.

² Biju Pattnaik University of Technology, Rourkele, 769015 India.

{dgjadhav2007@yahoo.co.in; shyampattnaik@yahoo.com}

Abstract. Memetic Algorithm (MA) is a metaheuristic search method. In proposed memetic algorithm two memes are used in definite proportion of refinement process with random selection. Genetic Algorithm due to its good exploration capability is used as framework algorithm and Swine Influenza Inspired Optimization (SIIO) as well as chemotaxis mechanism of Bacterial Foraging Optimization (BFO) are used as memes. The memetic process is realized using global best fitness among individual health information in SIIO and by imitating the nutrient information from the bacteria of the best fitness in BFO. The proposed variant of memetic algorithm is applied on the standard benchmark functions with modal variations, discontinuity and noise properties. When the results are compared, the proposed memetic algorithm shows better performance than MA using SIIO (sMA) and MA using BFO (bMA) both in terms of speed of convergence and quality of solutions.

Keywords: Hyper-heuristic. multi-meme. Memetic Algorithm (MA). Swine Influenza Inspired Optimization (SIIO). Bacterial Foraging Optimization (BFO).

1 Introduction

Application of meme in Genetic Algorithm (GA) is known as Memetic Algorithms (MAs) [1]. GA having the meme embedded in it has been introduced as Memetic Algorithm (MA) [1,2]. It is inspired by both Darwinian principle of natural evolution and Dawkins' notion of a meme as a unit of information or idea transmission from one to another or cultural evolution capable of individual learning [1]. In general, MA is a synergy of evolution and individual learning [1], which improving the capability of evolutionary algorithms like GA for finding optimal solutions in function optimization problems with accuracy and convergence speed.

Stochastic optimizers widely used are Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Bacterial Foraging Optimization (BFO), Differential Evolution (DE), Evolutionary Programming (EP) etc. [3, 4, 5, 6, 7] where a population of solutions is utilized in the search process. The Swine Influenza inspired Optimization (SIIO) and Swine Influenza Model Based Optimization (SIMBO) are newly introduced algorithms based on the epidemic outbreak and viral spread [8, 9]. These algorithms are able to explore and exploit the promising regions in the search space but take relatively more time [1]. Hence, algorithms are hybridized for utilizing better exploration and exploitation capacities for making this algorithm faster as well as accurate. Hybridization of GA with BFO is done for global optimization [10]; and also for exploration of search space in obtaining the local minima of the present location, the simplex algorithm of Nelder-Mead is used along with BFO [11]. A Real-Coded Memetic Algorithm based on crossover-hill climbing maintains a pair of parents and performs crossover until offspring are obtained for the best offspring selection [2]. In developing some of the memetic algorithms, GA is used and gradient-based information is utilized as meme [12]. But the gradient-based methods are failing in the cases of multimodal and non-differentiable functions [12]. Heuristic search algorithms are beneficial over the gradient type search algorithms for overcoming local optima trapping. Considering this property of heuristic search method, it is used for local refinement in the memetic algorithm [13]. One such combination of PSO and stochastic local search is made for handling multimodality of functions [14]. The population based refinement mechanism searches the small area for minima as well as takes out the solution from local trapping to go to better fitness landscape nearby [13][15]. Similarly, in this paper Memetic Algorithm (MA) is proposed by hybridization of GA with SIIO as meme for this new MA and also GA with BFO as meme is available in [13][15] is utilized. Also the use of multi-meme has advantages from all memes [16, 17], so in proposed work optimizing nature of individual's health information spread in SIIO and random search in nutrient gradients of BFO's chemotaxis mechanism both are simultaneously utilized.

This paper addresses two issues, utilization of SIIO as meme along with GA in making MA and use of two memes in conjunction with GA in the memetic algorithm to make multi-meme memetic algorithms. Swine Influenza Inspired Optimization (SIIO) is newly introduced population based algorithm. SIIO as well as BFO's chemotaxis mechanism both are used as memes for refinement in the same algorithm. The motivation of using different memes is to facilitate competition and cooperation in the multiple memes used in the memetic search for more effectiveness and efficiency. The paper is organized as follows. Section 2 presents the proposed Memetic Algorithm. Section 3 is for experimentation and result discussion. Finally section 4 concludes the paper.

2 Memetic Algorithm using multi-meme

GA represents an intelligent exploration with a random search confined in a predefined search space for solving a problem optimally with the help of population [3]. It is having evolution in an iterative way and generation of new population (offspring) from old one. All genes are in form of a string and are the encoded binary, real etc. version of a candidate solution [2, 3]. An evaluation function (fitness function) is used to assign a fitness value to every string, showing its fitness for the problem. Standard GA uses genetic operators such as selection, crossover, and mutation on initially generated random population in order to evolve a complete generation of new strings [3]. GA is having the capability to handle complex problems involving features such as discontinuities, multimodality and disjoint feasible spaces [3]. Elitism prevents the loss of good solutions once they are detected and also speeds up the performance of the GA significantly [3]. These best solutions are used in the reproduction process, which helps the algorithm convergence.

The unique aspect of MA is that the chromosomes are facilitated to gain some experience with a refinement process in between regular evolutionary process [1]. Similar to the GA, MA also generates an initial population randomly and searches in the fitness landscape. Subsequently, the refinement process leads solutions in the direction of close optima. These improvements are going to accumulate over all generations, resulting in a larger improvement in total performance [1][13]. The hyper-heuristic search mechanism in the memetic algorithms offers the speed and quality of convergence [16, 17].

This paper uses Genetic Algorithm as framework and the SIIO as well as BFO's chemotaxis mechanism are used as the memes, where both the meme are heuristic. The selection of either SIIO or BFO's chemotaxis mechanism is random for meme application, making algorithm hyperheuristic. It is having the advantages by facilitating a chance to multi-memes for improving chromosome in the MA search. The memes are applied with the proportion of SIIO ranging from 95% to 60% while BFO's chemotaxis mechanism with 5% to 40%.

The population is used as real coded for algorithmic framework and meme process application. The framework has the population which is obtained by generation process. The generation process generates the uniformly distributed random population for particular function as presented in Table 1, where 'F' is function, 'D' is dimension and 'NFFE' is number of fitness function evaluations. The fitness of the population is determined and the current best individual is found out for the elite preservation. Then this population goes through the GA operations like crossover and mutation. The crossovers are used as BLX- α crossover [18] and SBX crossover [19]. The offspring $O^1 = \{o^1_1, o^1_2, \dots, o^1_i, \dots, o^1_d\}$ and $O^2 = \{o^2_1, o^2_2, \dots, o^2_i, \dots, o^2_d\}$ are generated from parents $P^1 = \{p^1_1, p^1_2, \dots, p^1_i, \dots, p^1_d\}$ and $P^2 = \{p^2_1, p^2_2, \dots, p^2_i, \dots, p^2_d\}$ with 'd' dimension. In BLX- α crossover O is chosen randomly between the interval $[P^1 - I \cdot \alpha), (P^2 + I \cdot \alpha)]$ with the condition $P^1 < P^2$ and $I = \max(P^1, P^2) - \min(P^1, P^2)$. In SBX crossover the effect of the one-point crossover of the binary representation is tried to emulate. The crossover generates two offspring $O^1 = 1/2[(1+B) P^1 + (1-B) P^2]$ and $O^2 = 1/2[(1-B) P^1 + (1+B) P^2]$ where $B \geq 0$ is a sample from random generator having density function as in (1).

$$p(B) = \frac{1}{2}(\eta+1)B^\eta, \quad \text{if } 0 \leq B \leq 1 \quad \frac{1}{2}(\eta+1) \frac{1}{B^{\eta+2}}, \quad \text{if } B > 1 \quad (1)$$

This distribution is obtained by using uniform random number source $u(0,1)$ as per (2).

$$B(u) = (2u)^{\frac{1}{\eta+1}}, \quad \text{if } u \leq \frac{1}{2} \quad \left(\frac{1}{2(1-u)}\right)^{\frac{1}{\eta+1}}, \quad \text{if } u > \frac{1}{2} \quad (2)$$

where $\eta=2$ is set for better results. Exploitation capacity of operator increases with higher values of η . The mutation is also done of the variables by changing individual randomly under uniform normal distribution. The meme is evoked after a number of generations of framework algorithm which is empirically selected as 2. In this MA, the fine searching of the search space is done around best candidate solution by process of exploitation. The small

population for the refinement of the current best individual is generated by perturbation. The choice of meme is random.

2.1 Multi-meme by SIIO or BFO's chemotaxis mechanism

The refinement of the best individual is used as meme process and the random meme is either SIIO or BFO. Swine flu spreads same as the regular seasonal flu [20, 21, 22]. The swine flu is having the speed in spread of infection and instating capacity also.

Early in the 20th century Kermack and McKendrick [20] [21] developed the first mathematical model describing an influenza epidemic. The basic model for this is known as the Susceptible-Infectious-Recovered (SIR) model [21] and is shown in Fig. 1. In this, the initial population is divided into the classes: Susceptible (S), Infectious (I), and Recovered (R). A susceptible class is of persons having acute febrile respiratory illness of swine influenza-A (H1N1) virus infection with onset within 7 days of close contact with a person who is a confirmed case of swine influenza A (H1N1) i.e. infectious class. Case of swine influenza A (H1N1) virus infection is confirmed with medical laboratory confirmation test. Depending on the test of confirmation of infection, individuals proceed from class S i.e. Susceptible to class I i.e. Infectious [21]. After the proper treatment with recommended medicinal dose, the recovered individual moves to class R i.e. Recovered, at that stage the individual acquires immunity to future infection. The researchers extended SIR model to represent and/or predict the spatial dynamics of an influenza epidemic. Rvachev devised the first spatio-temporal model of influenza in the 1960s [22]. Later researchers modified the basic SIR model by including seasonality, vaccination, treatment etc [20]. The SIR model extended with incorporation of vaccination and treatment [20] is shown in Fig. 2. The individuals from class S can be vaccinated and infected individuals from class I are treated with antiviral drugs. During the treatment phase they are transferred into the class T [20]. Individuals from class S who are vaccinated are taken into class V and they are considered immune. After the treatment individuals from class T goes to class R due to recovery [20].

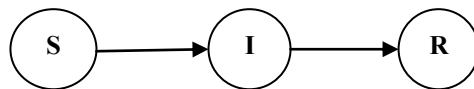


Fig. 1. Basic Susceptible-Infectious-Recovered model of disease transmission

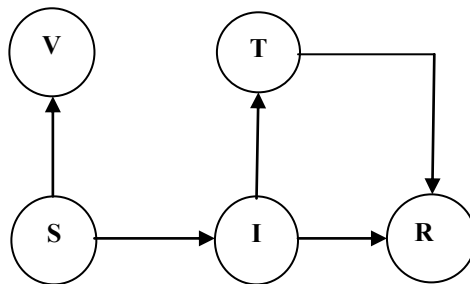


Fig. 2. Extended Susceptible-Infectious-Recovered Model

The SIIO is a new optimization technique, which is based on the extended SIR model [8]. It shows heuristic optimization through vaccination and treatment. As mechanism of spreading the infection among the population is arbitrary, the mechanism of the control of the epidemic is also arbitrary.

This concept is utilized to frame optimization algorithm with spread and control of the swine flu virus in the population because every transition in the classes is arbitrary. These transitions are considered as optimization operations or happen similar to optimization steps. In these transitions or state change of individuals, reference is the best individual. The fitness value is taken as indication to decide the tag of the class for individuals and individual having best fitness value is used for spreading health among the individuals. Other individuals try to enhance their health by imitating the health information of the best individual like preventive care practiced, medicinal information etc. The individuals having best health are preserved during the process of optimization. In this mechanism poor fitness ill individual tries to drag the population towards the viral spread whereas healthier ones towards better health. The optimization model SIIO is presented in Fig. 3.

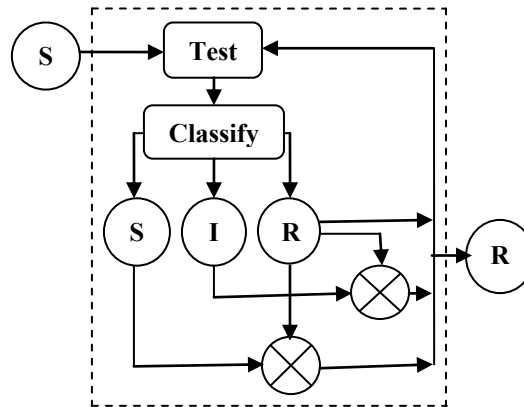


Fig. 3. Model of the SIIO [8]

The susceptible and infected individuals are quarantined as per (3). Quarantine is based on threshold given in (4).

$$P(t+1) = P(t) \times (1.5 \times \text{rand} - 0.25) \quad (3)$$

$$DT = CF(2) - 0.05 \times (CF(2) - CF(1)) \quad (4)$$

$P(t+1)$ is new individual value calculated by changing the current value $P(t)$. DT is dynamic threshold and CF is current fitness or current health of individual. $CF(1)$ is current best individual health and $CF(2)$ is next better fitness. The treatment is given to individuals in population with varying amount of antiviral drugs. The dose of antiviral drugs is dependent on current health, pandemic health and changes adaptively based on the difference in the current individual value and best individual value. As the health of individual varies, dose quantity also changes as given in (5) and individual treatment is done by (6).

$$\text{Dose} = c \times \text{Dose} + \text{rand} \times (P(t) - P(1)) \quad (5)$$

$$P(t+1) = P(t) + \text{Dose} \quad (6)$$

$P(t)$ is current individual value and $P(1)$ is best individual value. $P(t+1)$ is new individual value calculated by changing the current value $P(t)$. The SIIO performs optimization in four steps as presented below. These steps continue for all generations.

Steps in the SIIO [8]

- Step 1- Evaluation of health of individuals by fitness function.
- Step 2- Classify individuals into S, I and R.
- Step 3- Quarantine of individual.
- Step 4- Treatment of Individual by dose.

Table 1. Benchmark numerical functions used for experiment [7, 8][15][17].

F	Function Name	Range	NFFE
F1	Sphere	$[-100, 100]^D$	150000
F2	Schwefel's 2.22	$[-10, 10]^D$	200000
F3	Schwefel's 1.2	$[-100, 100]^D$	500000
F4	Schwefel's 2.21	$[-100, 100]^D$	500000
F5	Rosenbrock	$[-30, 30]^D$	500000
F6	Step	$[-100, 100]^D$	150000
F7	Quartic	$[-1.28, 1.28]^D$	300000
F8	Schwefel's 2.26	$[-500, 500]^D$	300000
F9	Rastrigin	$[-5.12, 5.12]^D$	300000
F10	Ackley	$[-32, 32]^D$	150000
F11	Griewank	$[-600, 600]^D$	300000
F12	Penalized #1	$[-50, 50]^D$	150000
F13	Penalized #2	$[-50, 50]^D$	150000

Pseudo code SIIO as meme application [8]

```

Initialize parameters: G, IN, D, Pq, Pt, Dose, P (D x IN)
for i=1:G
    for j = 1 : IN
        CF (j) =Evaluate Fitness Function
    end
    Update P and PH
    Sort individual in order of ascending health
     $DT = CF(2) - 0.05 \times (CF(2) - CF(1))$ 
    for k=1: IN
        if k==1
            tag individual as best health (R)
        elseif k<IN/2
            tag individual as suspected (S)
        else
            tag individual as infected (I)
        end
        if (CF(k)-DT)>0 & rand< Pq & P(k)  $\subset$  S,I
            for l=1:D
                P(l,k)=P(1:k) x (1.5 x rand-0.25)
            end
        end
        if rand < Pt & P(k)  $\subset$  S
            Dose=c x Dose + rand x (P(k)-P(1))
            P(k)=P(k) + Dose
        end
    end
end

```

where G is number of generations, IN size of population, D is dimension of population, P_q is probability of quarantine, P_t is probability of treatment, Dose is amount of dose for treatment, P (D x IN) is population of size D by IN and CF is current fitness.

In case of SIIO as in refining process, initially when error is above 0.01 the search intensity is tuned higher and when it is below 0.01, intensity is decreased by about 75% of the previous one.

In BFO, an E. coli bacterium is alternating into run and tumble actions [5][10] i.e. two processes for climbing nutrient gradients. So in the bacterial foraging optimization, the gradient information and the heuristic are combined for the search. In the proposed Memetic Algorithm chemotaxis mechanism of BFO is used as one of the memes.

The step size C(i) is set based on the fitness value (J). For initial search with higher fitness value up to 1, the step size is used as 0.1. When the fitness value is lower i.e. up to 10^{-5} the step size is set equal to fitness value. As fitness value goes below 10^{-5} , step size is used is little higher than it as indicated. This prevents bacteria from locking, oscillations and is giving them mobility. This makes the algorithm more exploiting in nature, suitable for refinement process.

Pseudo code of BFO's Chemotaxis as meme application

Initialize P, S, Nc, Ns, Nre, and C (i)

1) Chemotaxis loop: j = j + 1

a) **for** i = 1, 2, ... S, take a chemotaxis step for bacterium i as follows;

b) Compute J(i, j, k, l). Where J is fitness, l is Elimination-dispersal loop counter and k is Reproduction loop counter; l and k are not used in chemotaxis.

c) Let Jlast = J(i, j, k, l) save J, can find better cost by run.

d) Set stepsize C(i) based on fitness (J)

if $l < J_{last}$,	C(i)= 0.1;	endif
if $10^{-5} < J_{last} < 1$,	C(i)= Jlast;	endif
if $10^{-6} < J_{last} < 10^{-5}$,	C(i)= 0.5×10^{-4} ;	endif
if $10^{-7} < J_{last} < 10^{-6}$,	C(i)= 10^{-5} ;	endif

if Jlast<10⁻⁷, C(i)= 5*10⁻⁶; **endif**

e) Tumble: Generate a random vector $\Delta(i) \in R^n$ with each element $\Delta_m(i)$, $m = 1, 2, \dots, p$, a random number on $[-1, 1]$, where p is dimension. θ is bacterium value.

f) Move: Let

$$\theta^i(j+1, k, l) = \theta^i(j, k, l) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^T(i) \Delta(i)}}$$

This results in a step of size $C(i)$ in the direction of the tumble for bacterium i .

g) Compute $J(i, j+1, k, l)$

h) Swim:

Let $m = 0$ (counter for swim length).

while $m < N_s$; N_s is swim length (if have not climbed down too long)

Let $m = m + 1$.

if $J(i, j+1, k, l) < J_{last}$ (if doing better),

let $J_{last} = J(i, j+1, k, l)$ and

$$\theta^i(j+1, k, l) = \theta^i(j+1, k, l) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^T(i) \Delta(i)}}$$

use $\theta(i, j+1, k, l)$ to compute new $J(i, j+1, k, l)$ as in (g).

else, let $m = N_s$, break (This is end of while statement.)

endif

endwhile

i) Go to next bacterium $(i+1)$ if $i \neq S$ (i.e., go to b) to process the next bacterium).

end for

2) **if** $j < N_c$, go to step 1. N_c is chemotactic counter. Continue chemotaxis - life of the bacteria is not over.

endif

In case of BFO as meme, the refinement intensity is tuned for better result. After the refinement by any of the meme application, the better result of the refinement process is accepted in framework algorithmic process. It strengthens the algorithm for convergence of various types of functions like unimodal, multimodal, discontinuous, noisy etc. functions. The performance of the algorithms is checked with 13 standard benchmark numerical functions presented in Table 1.

Pseudo Code of proposed MA

Initialization: Generate a random initial population

Number of generation: $G = 0$, $G_{meme} = 0$

while Stopping conditions are not satisfied **do**

Evaluate all individuals in the population

Find current best individual

if $G_{meme} = 2$ **do** (refinement by BFO or SIIO)

Obtain small population for the best individual by perturbation

while stopping conditions are not satisfied **do**

if BFO is selected **do** (apply BFO)

Take a chemotaxis step for bacterium

Compute $J(i, j, k, l)$; save J (run may find better cost)

Tumble: in a step of size $C(i)$

```

    Compute  $J(i, j + 1, k, l)$  i.e. cost
    Set  $C(i)$  step size as per criterion
    Swim: with counter  $N_s$ 
    if  $J(i, j + 1, k, l) < J_{last}$  continue swim
    else stop swim
    endif
    Evaluate and sort the population
else (apply SIIO)
    Evaluate the population
    Classify individuals into S, I and R
    Quarantine of individual
    Treatment of Individual by dose
endif
endwhile
    Gmeme=0
    Return best individual to main algorithm
endif
    Select individuals for crossover
    Crossover by BLX- $\alpha$  or SBX crossover operators with probability  $P_c$ 
    Correct the feasibility of the produced individuals (offspring)
    Mutate some of the descendant population using mutation probability  $P_m$ 
    Replace the old population by offspring preserving the elite
    G++, Gmeme++
end while

```

3 Experimentation

GA is framework algorithm that is exploring the promising regions and then refining process is being carried out by either BFO or SIIO. In all the algorithms, a population of 100 individuals is used having real-valued representation of 10 dimensions. The crossover operator is used as BLX- α crossover or SBX crossover with crossover probability set to 0.8. Parameters are used for BLX- α crossover, $\alpha = 0.3$ and for SBX crossover, $\eta = 2$. Mutation probability is set as 0.05. The crossover probability, mutation probability and other parameters are set for the better results. The algorithm is using generational replacement of individuals by preservation of one elite. Either number of fitness function evaluations (NFFE) as shown in Table 1 or error in fitness value of 10^{-8} is considered as stopping criterion.

Algorithms are run for 25 trails for testing the consistency of the convergence and their results are presented in Table 2. For comparison purpose MA with only SIIO as meme (sMA-BLX and sMA-SBX), MA with only BFO as meme (bMA-BLX and bMA-SBX)[15] and proposed mMMA having both SIIO and BFO as memes (mMA-BLX and mMA-SBX) are implemented with same parameters i.e. dimensions, population, stopping criterion etc. Proposed mMMA has the meme selection by random selection process with higher selection rate for SIIO than that for BFO. Here SIIO and BFO are selected with 95% to 60% proportion for SIIO and 5% to 40% proportion for BFO.

Table 2. mean(std dev) of best fitness error of sMA, bMA and Proposed mMMA

Algorithm	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13
sMA-BLX	1.04E-9 (1.32E-9)	3.06E-9 (2.58E-9)	1.45E-9 (2.49E-9)	2.46E-9 (2.73E-9)	6.99E+0 (1.70E-1)	0.00E+0 (0.00E+0)	3.28E-1 (2.22E-1)	1.54E-7 (4.60E-7)	5.38E-10 (9.06E-10)	3.26E-9 (2.75E-9)	1.15E-9 (1.75E-9)	9.12E-2 (5.17E-2)	9.21E-2 (2.84E-2)
sMA-SBX	8.9E-10 (1.24E-9)	3.13E-9 (2.60E-9)	2.47E-9 (2.90E-9)	2.96E-9 (1.90E-9)	6.98E+0 (2.49E-1)	0.00E+0 (0.00E+0)	3.40E-1 (2.29E-1)	4.01E-8 (2.82E-7)	1.42E-9 (1.86E-9)	3.14E-9 (2.59E-9)	1.94E-9 (2.98E-9)	2.69E-1 (1.73E-1)	1.00E-1 (2.64E-2)
bMA-BLX [15]	6.86E-9 (1.96E-9)	1.69E-2 (5.74E-3)	8.30E-9 (1.22E-9)	2.86E-3 (3.17E-4)	5.75E+1 (1.31E+2)	1.68E+2 (1.42E+2)	6.00E-1 (2.99E-1)	4.94E+2 (3.97E+2)	2.21E+1 (5.96E+0)	4.78E-3 (5.50E-4)	4.04E+0 (2.34E+0)	3.72E+0 (3.96E+0)	1.12E-2 (7.17E-3)
bMA-SBX [15]	7.31E-9 (1.87E-9)	1.99E+0 (9.89E+0)	8.54E-9 (1.14E-9)	2.54E-3 (3.79E-4)	5.86E+1 (1.03E+2)	1.44E+3 (8.66E+2)	5.10E-1 (2.54E-1)	6.81E+2 (3.68E+2)	3.25E+1 (1.07E+1)	2.32E+0 (4.02E+0)	6.43E+0 (3.81E+0)	8.95E+0 (4.95E+0)	6.07E-2 (1.79E-1)
mMA-BLX s95b05	5.07E-10 (1.13E-9)	2.47E-9 (2.40E-9)	1.20E-9 (2.28E-9)	2.99E-9 (2.46E-9)	6.99E+0 (1.88E-1)	0.00E+0 (0.00E+0)	2.67E-1 (2.26E-1)	1.64E-4 (3.68E-3)	9.92E-10 (2.38E-9)	3.14E-9 (3.18E-9)	1.56E-9 (2.04E-9)	1.74E-2 (5.62E-3)	1.22E-2 (1.10E-2)
mMA-SBX s95b05	2.45E-10 (5.05E-10)	3.20E-9 (3.01E-9)	2.23E-9 (2.64E-9)	2.95E-9 (2.22E-9)	6.98E+0 (2.59E-1)	0.00E+0 (0.00E+0)	3.15E-1 (2.51E-1)	5.81E-5 (2.92E-4)	6.31E-10 (1.21E-9)	3.42E-9 (3.10E-9)	8.44E-10 (1.81E-9)	1.75E-2 (5.14E-3)	1.32E-2 (8.81E-3)
mMA-BLX s90b10	4.06E-10 (7.34E-10)	1.99E-9 (2.22E-9)	2.44E-9 (2.94E-9)	2.52E-9 (2.47E-9)	7.00E+0 (2.41E-1)	0.00E+0 (0.00E+0)	3.00E-1 (2.28E-1)	5.60E-6 (2.80E-5)	1.43E-9 (2.07E-9)	3.16E-9 (2.72E-9)	1.62E-9 (2.69E-9)	7.25E-3 (4.77E-3)	1.71E-2 (1.25E-2)
mMA-SBX s90b10	1.05E-9 (2.52E-9)	2.98E-9 (3.08E-9)	1.21E-9 (1.47E-9)	2.72E-9 (2.46E-9)	7.06E+0 (1.12E-1)	0.00E+0 (0.00E+0)	2.88E-1 (2.19E-1)	2.01E-4 (1.24E-3)	1.14E-9 (2.00E-9)	3.15E-9 (2.47E-9)	1.40E-9 (2.16E-9)	8.46E-3 (3.89E-3)	1.26E-2 (1.05E-2)
mMA-BLX s80b20	9.57E-10 (1.93E-9)	2.67E-9 (2.33E-9)	2.77E-9 (2.91E-9)	2.52E-9 (2.16E-9)	7.01E+0 (1.82E-1)	0.00E+0 (0.00E+0)	3.59E-1 (2.85E-1)	1.87E-3 (1.12E-2)	6.37E-10 (1.13E-9)	3.29E-9 (3.00E-9)	1.90E-9 (3.19E-9)	2.94E-4 (1.70E-4)	1.17E-2 (1.09E-2)
mMA-SBX s80b20	1.00E-9 (2.45E-9)	1.83E-9 (1.82E-9)	1.55E-9 (2.44E-9)	3.30E-9 (2.96E-9)	7.08E+0 (1.56E-1)	0.00E+0 (0.00E+0)	2.94E-1 (2.66E-1)	3.40E-5 (1.73E-4)	1.36E-9 (2.15E-9)	2.94E-9 (2.69E-9)	1.12E-9 (2.10E-9)	3.02E-4 (1.06E-4)	9.77E-3 (6.38E-3)
mMA-BLX s70b30	8.41E-10 (1.75E-9)	2.73E-9 (2.88E-9)	2.19E-9 (2.69E-9)	3.08E-9 (3.04E-9)	7.02E+0 (2.61E-1)	0.00E+0 (0.00E+0)	2.75E-1 (2.60E-1)	1.03E-2 (4.23E-2)	1.15E-9 (1.47E-9)	1.74E-9 (2.63E-9)	1.97E-9 (3.04E-9)	1.28E-4 (3.92E-5)	9.41E-3 (1.19E-2)
mMA-SBX s70b30	6.15E-10 (1.61E-9)	3.14E-9 (2.50E-9)	1.82E-9 (2.17E-9)	2.70E-9 (2.93E-9)	7.11E+0 (1.28E-1)	0.00E+0 (0.00E+0)	3.04E-1 (2.12E-1)	1.34E-6 (6.14E-6)	9.55E-10 (1.77E-9)	2.44E-9 (2.49E-9)	1.15E-9 (1.82E-9)	1.31E-4 (3.49E-5)	8.00E-3 (6.49E-3)
mMA-BLX s60b40	5.16E-10 (1.06E-9)	3.19E-9 (3.15E-9)	2.76E-9 (3.59E-9)	3.96E-9 (2.94E-9)	7.12E+0 (1.13E-1)	0.00E+0 (0.00E+0)	3.13E-1 (2.52E-1)	3.66E-3 (1.88E-2)	8.90E-10 (1.40E-9)	3.43E-9 (2.96E-9)	8.08E-10 (1.71E-9)	8.45E-5 (1.85E-5)	1.02E-2 (8.22E-3)
mMA-SBX s60b40	8.24E-10 (1.71E-9)	3.35E-9 (2.81E-9)	2.28E-9 (2.31E-9)	2.10E-9 (2.47E-9)	7.12E+0 (1.19E-1)	0.00E+0 (0.00E+0)	4.34E-1 (2.36E-1)	5.72E-4 (7.44E-3)	1.35E-9 (2.34E-9)	4.00E-9 (3.01E-9)	1.30E-9 (2.03E-9)	8.66E-5 (1.76E-5)	1.22E-2 (1.29E-2)

3.1 Discussion

The algorithms are tested on standard benchmark numerical functions with unimodal properties (functions F1-F5), multimodal properties (function F8-F13) as well as noise property with uniformly distributed random noise (function F7) and discontinuity property (function F6) [7, 8][15][17]. Functions F8-F13 are multimodal functions in which the number of local minima is increasing exponentially with the problem dimension [7]. They are the most difficult class of problems for many optimization algorithms [7]. For unimodal functions, the convergence rates are more interesting than the final results of optimization. In case of multimodal functions, the final results are important since they reflect an algorithm's ability of escaping poor local optima. Table 2 presents the performance of all test functions in terms of mean and standard deviation (Std Dev) of the best error in near optimum in 25 trails, where 'F' is function. The proposed MA converges successfully for many unimodal and multimodal functions and shows promising performance for other functions also. The convergence graphs in Fig.1 (a)-(h) show number of fitness function evaluations on x-axis and error in the optimum value on y-axis. For comparison the convergence characteristics of sMA, bMA, and proposed mMMA are plotted in the graphs. The mMMA with BLX- α and SBX operators are performing in similar manner.

For functions F1 and F3 all MAs (bMA, sMA and mMMA) are converging successfully by attaining fitness error of 10^{-8} . For functions F2, F4, F6 and F9-F11 all MAs are converging successfully except bMA by attaining fitness error of 10^{-8} . The sMA and proposed mMMA converges for function F8 for many runs. For functions F5-F9, F11-F13 exhibits improvement in convergence behavior. In these cases sMA is outperforming as seen from the graphs. Algorithm sMA is having optimizing nature using individual's health information spread in refinement process and performing better in unimodal and some multimodal functions but random refinement process mechanism of sMA also is beneficial in unimodal functions as well as multimodal functions when used to support for multi-meme algorithm.

4 Conclusion

In this paper a variant of memetic algorithm is developed. First, Swine Influenza Inspired Optimization (SIIO) is used as meme in development of Memetic Algorithm. Then two memes SIIO and BFO are utilized in developing multi-meme Memetic Algorithm. This Memetic Algorithm performs more robustly in terms of accuracy and convergence speed as compared to Memetic Algorithms those use only one type of meme, either SIIO or BFO. All the algorithms are attaining convergence accuracy up to the error of 10^{-8} for some of the functions and show better speed, success and consistency of the convergence. For rest of the functions also they are showing promising performance. The multimeme MA shows better performance as compared to both single meme MAs and also converges for more number of functions by taking the advantages from both the memes.

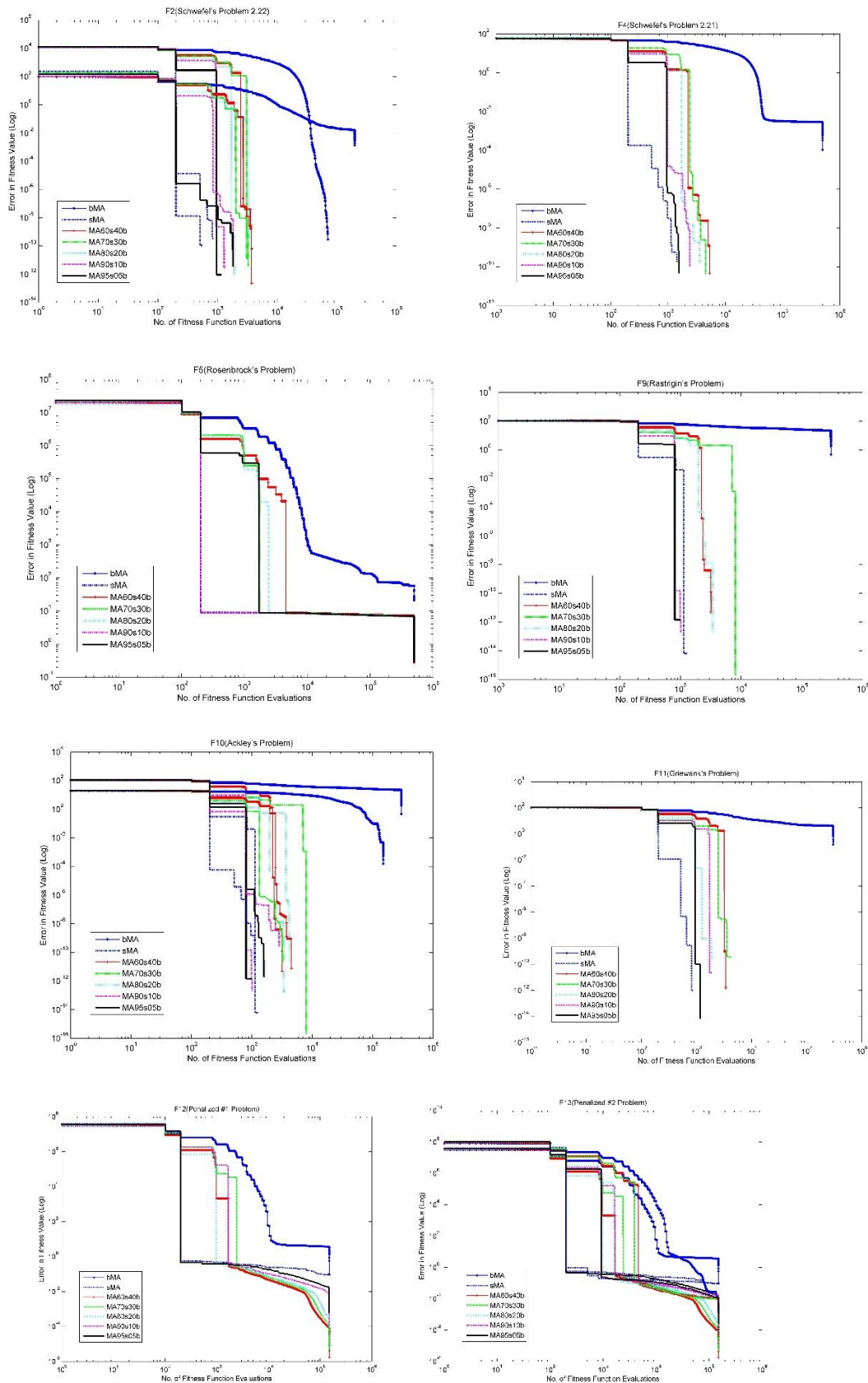


Fig. 1. Fitness Error versus number of function evaluation calls of test functions averaged over 25 runs
a) F2, b) F4, c) F5, d) F9, e) F10, f) F11, g) F12 and h) F13.

References

- [1] Nguyen, Q.H., Ong, Y.S., Krasnogor, N.: A Study on the Design Issues of Memetic Algorithm. In Proc.: IEEE Congr. Evol. Comput. pp. 2390-2397.(2007)
- [2] Lozano, M., Herrera, F., Krasnogor, N., Molina, D.: Real-Coded Memetic Algorithms with Crossover Hill-Climbing. *Evol. Comput.* 12(3), 273-302 (2004)
- [3] Michalewicz, Z.: *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag. (1996)
- [4] Liang, J.J., Qin, A.K., Suganthan, P.N., Baskar, S.: Comprehensive Learning Particle Swarm Optimizer for Global Optimization of Multimodal Functions. *IEEE Trans. Evol. Comput.* 10(3), 281-295 (2006)
- [5] Passino, K.M. : Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control. Syst. Mag.* pp. 52-67 (2002)
- [6] Das S., Suganthan, P. N.: Differential evolution - a survey of the state-of-the-art. *IEEE Trans. Evol. Comput.* 15(2), 04-31 (2011)
- [7] Yao, X., Liu, Y., Lin, G.: Evolutionary Programming Made Faster. *IEEE Trans. Evol. Comput.* 3(2), 82-102 (1999)
- [8] Pattnaik, S.S., Jadhav, D.G., Devi, S., Ratho, R.K.: Swine influenza inspired optimization algorithm and its application to multimodal function optimization and noise removal. *Artificial Intelligence Research.* 1(1), 18-30 (2012) (DOI:10.5430/air.v1n1p18)
- [9] Pattnaik, S.S., Bakwad, K.M., Sohi, B.S., Ratho, R.K., Devi, S.: Swine influenza models based optimization (SIMBO). *Applied Soft Computing.* 13(1), 628-653 (2013)
- [10] Kim, D.H., Abraham, A., Cho, J.H.: A hybrid genetic algorithm and bacterial foraging approach for global optimization. *Inform. Sci.* 177(18), 3918-3937 (2007)
- [11] Panigrahi B.K., Pandi, V.R.: Bacterial foraging optimisation: Nelder–Mead hybrid algorithm for economic load dispatch. *IET Gener. Transm. Distrib.* 2(4), 556-565 (2008)
- [12] Li, B., Ong, Y.-S., Le, M.N., Goh , C.K.: Memetic Gradient Search. In Proc.: IEEE Congr. Evol. Comput. pp. 2894-2901 (2008)
- [13] Jadhav, D.G., Pattnaik, S.S., Devi, S., Lohokare, M.R., Bakwad, K.M.: Approximate Memetic Algorithm for Consistent Convergence. In Proc.: National Conference on Computational Instrumentation. pp. 118-122. Chandigarh (2010)
- [14] Akbari, R., Ziarati, K.: Combination of Particle Swarm Optimization and Stochastic Local Search for Multimodal Function Optimization. In Proc.: IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application. pp. 388-392 (2008)
- [15] Devi, S., Jadhav, D.G., Pattnaik, S.S., : Memetic Algorithm and its application to Function Optimization and Noise removal. In Proc.: IEEE World Congr. Inform. Commun. Technologies. pp.748-753. Mumbai (2011)
- [16] Meuth, R., Lim, M.-H., Ong, Y.-S., Wunsch II, D.C.: A proposition on memes and meta-memes in computing for higher-order learning. *Memetic Computing.* 1(2), 85-100 (2009)
- [17] Jadhav, D.G., Devi, S., Pattnaik, S.S.: Memetic Algorithm with PSO or BFO's Chemotaxis Mechanism as Local Search for Function Optimization. *International Conference on Recent Advances in Computing and Software Systems.* pp. 42-47. Chennai (2012)
- [18] Eshelman L.J., Schaffer, J.D.: Real-coded genetic algorithms and interval-schemata. In: Darrell, W.L. (ed.) *Foundation of genetic algorithms*, vol. 2, pp. 187–202. Morgan Kaufmann, San Mateo (1993)
- [19] Deb, K., Agrawal, R.B.: Simulated binary crossover for continuous search space. *Complex Syst.* 9(2), 115-148 (1995)
- [20] Coburn, B.J., Wagner, B.G., Blower, S.: Modeling influenza epidemics and pandemics: insights into the future of swine flu (H1N1). *BMC Medicine.* 7-30 (2009) (doi:10.1186/1741-7015-7-30)
- [21] Kermack, W.O., McKendrick, A.G.: A contribution to the mathematical theory of epidemics. In Proc.: *Roy. Soc.*, vol.115, pp. 700-721. London (1927)
- [22] Rvachev, L.A.: Modeling experiment of a large-scale epidemic by means of a computer. (In Russian.) *Trans. USSR Acad. Sci. Ser. Mathematics and Physics.* 180(2), 294-296 (1968)