Shahid Bahonar University of Kerman
Journal of Mahani Mathematical Research
Print ISSN: 2251-7952 Online ISSN: 2645-4505

A NEW IMPROVED FRUIT FLY OPTIMIZATION ALGORITHM BASED ON PARTICLE SWARM OPTIMIZATION ALGORITHM FOR FUNCTION OPTIMIZATION PROBLEMS

R. Etesami, M. Madadi ✉, and F. Keynia

Article type: Research Article

(Received: 14 November 2022, Received in revised form 01 October 2023)

(Accepted: 11 November 2023, Published Online: 18 November 2023)

Abstract. The Fruit Fly Optimization algorithm is an intelligent optimization algorithm. To improve accuracy, convergence speed, as well as jumping out of local optimum, a modified Fruit Fly Optimization algorithm (MFFOV) is proposed in this paper. The proposed algorithm uses velocity in particle swarm optimization and improves smell based on dimension and random perturbations. As a result of testing ten benchmark functions, the convergence speed and accuracy are clearly improved in Modified Fruit Fly Optimization (MFFOV) compared to algorithms of Fruit Fly Optimization (FFO), Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC), Teaching-Learning-Based Optimization (TLBO), Genetic Algorithms (GA), Gravitational Search Algorithms (GSA), Differential Evaluations (DEs) and Hunter–Prey Optimizations (HPOs). A performance verification algorithm is also proposed and applied to two engineering problems. Test functions and engineering problems were successfully solved by the proposed algorithm.

Keywords: Fruit fly optimization algorithm, particle swarm optimization, random perturbation, velocity.
2020 MSC: 65K10.

1. Introduction

In the definition of optimization, it can be said that optimization refers to the process of finding optimal values for the parameters of a system from all possible values to maximize or minimize its output [20]. Problems in various disciplines can be turned into optimization problems, and a method to solve them can be designed accordingly. This reveals the importance of different optimization methods and provides an exciting research path for researchers [13]. In recent years, intelligent optimization algorithms have been considered by many scholars. These algorithms are simple and effective search methods, and they are mostly used in optimization problems. Some of them include: Particle Swarm Optimization (PSO) [16], Ant Colony Optimization (ACO) [9], Artificial

✉ madadi@uk.ac.ir, ORCID: 0000-0002-7950-138X

https://doi.org/10.22103/jmmr.2023.20538.1362
Publisher: Shahid Bahonar University of Kerman
How to cite: R. Etesami, M. Madadi, F. Keynia, *A new improved fruit fly optimization algorithm based on particle swarm optimization algorithm for function optimization problems*, J. Mahani Math. Res. 2024; 13(2): 73 - 91.

Bee Colony (ABC) [14], Fruit Fly Optimization (FFO) [26], Water Evaporation Optimization (WEO) [15], Symbiotic Organism Search (SOS) [2], Spider Monkey Optimization (SMO) [28], Lion Optimization Algorithm (LOA) [33], Golden Search Optimization algorithm (GSO) [24], Snake Optimizer (SO) [12], Dwarf Mongoose Optimization algorithm (DMO), and Hunter-prey optimizer (HPO) [23]. Particle Swarm Optimization algorithm was introduced by Eberhart and Kennedy in 1995. It was inspired by social behavior of bird flocking that are looking for food. Each solution is a particle and each particle adjusts its position in the search space according to the flying experience of its own and of its neighbors. PSO has shown a good performance in solving many of optimization problems. Also, the Fruit Fly Optimization algorithm is one of the intelligent optimization algorithms that was introduced by Taiwan scholar Wen-Tsao Pan in 2011. It is an optimization algorithm based on the food-finding behavior of fruit fly. Advantages of this method compared to other intelligent optimization algorithms include: it is simple, its adjustment parameters are less (only requires the adjustment of two parameters, namely, population Size and maximum generation number), its convergence speed is fast, and it's easy to implement [8]. Also, this algorithm has been successful for solving some problems such as the continuous function optimization [25], Travelling Salesperson Problem [14], Generalized Regression Neural Network parameter optimization [17], Optimization of industrial Structure configuration [32], Grey neural network parameter optimization [6], multidimensional knapsack problem [31] and so on. However, there are still some shortcomings, such as easily trapped into the local optimal value and low convergence accuracy. For this purpose, this paper proposes a Modified Fruit Fly Optimization Algorithm (MFFOV) based on velocity variable, Improve Smellbest based on dimension and random perturbation. The innovation of the article is as follows:

•Adding the speed variable to the FFO algorithm increases the exploration of the search and makes individuals move towards the food more accurately in each iteration.

•The random disturbance gives the misguided individuals more chances to find food.

•An important weakness of the FFO algorithm is its poor convergence in high dimensions. This problem is largely solved by examining each dimension in relation to the position of the fruit fly.

The remainder of this paper is structured as follows: Section 2 provides related works, A brief introduction of the fruit fly optimization algorithm is indicated in Section 3, and the Modified Fruit Fly Optimization Algorithm (MFFOV) is described in Section 4. Results and discussion are given in Section 5. In Section 6, the performance of the proposed algorithm for solving real-world problems was evaluated. And finally, a conclusion is presented in Section 7.

2. Related works

Fuqing and colleagues (2019) stated in a research article titled A two-stage differential biogeography-based optimization algorithm and its performance analysis: Biogeography-based optimization (BBO) has drawn a lot of attention for its outstanding performance. However, as with certain typical swarm optimization algorithms, BBO severely suffers from premature convergence problems and the rotational variance of the migration operator. In this paper, a two-stage differential biogeography-based optimization (TDBBO) is proposed to address the premature convergence problem and alleviate the rotational variance [29]. Li and colleagues (2020) stated in a research article titled Improved Fruit Fly Algorithm on Structural Optimization: To improve the efficiency of the structural optimization design in truss calculation, an improved fruit fly optimization algorithm was proposed for truss structure optimization. The fruit fly optimization algorithm was a novel swarm intelligence algorithm. In the standard fruit fly optimization algorithm, it is difficult to solve the high-dimensional nonlinear optimization problem and easy to fall into the local optimum. To overcome the shortcomings of the basic fruit fly optimization algorithm, the immune algorithm self-non-self-antigen recognition mechanism and the immune system learn-memory-forgetting knowledge processing mechanism were employed [18]. Darvish and colleagues (2018) stated in a research article titled Improved Fruit-Fly Optimization Algorithm and Its Applications in Antenna Arrays Synthesis: Synthesizing antenna arrays is one of the most influential optimization problems in the electromagnetics community. In this paper, an improved fruit-fly optimization algorithm (FOA) [entitled averager engine linear generation mechanism of candidate solution of FOA (AE-LGMS-FOA)] is proposed to be used in antenna array synthesis. This improvement includes adding a new search mechanism to enhance the efficiency of the algorithm for high-dimensional problems [7]. Bezdan and colleagues (2021) stated in a research articletitled Hybrid fruit-fly optimization algorithm with k-means for text document clustering: This study models the steelmaking rescheduling problem with flexible processing time as a complex hybrid flowshop in which two types of disruptions, machine breakdown and processing variation, are considered concurrently. A weighted sum of the five objectives, including minimization of the average sojourn time, earliness penalty, tardiness penalty, cast-break penalty, and system instability penalty, is considered in the proposed algorithm. We develop an effective hybrid fruit fly optimization algorithm (HFOA) that applies two vectors to represent individuals and presents routing and scheduling neighborhood structures [4].

3. Fruit Fly Optimization Algorithm

The fruit fly optimization algorithm is a method based on the food-finding behavior of the fruit fly for finding the global optimum. Compared to other

TABLE 1. Comparison of some metaheuristic algorithms with their advantages and disadvantages.

Ref	Proposed algorithm	Optimization criteria	Comparative methods	Advantages	Disadvantages
[35]	TDBBO Algorithm	One-objective optimization problems	BB, GBBO, PBBO, Algorithm	Solve the problem of early convergence and reduce rotational variance in the migration phase	Local optimization
[29]	IRRO-CSO Algorithm	Multiobjective and hybrid optimization problems	RRO, IRR, WOA, CSO, BAT, PSO Algorithms	Solve the problem of early convergence in the RRO algorithm and establish a balance between exploration and operation in the combined CSO IRRO algorithm	Early Convergence and entanglement in local optimality
[27]	CMVO Algorithm	Chaos theory in increasing local MVO search	ABC, MVO, MFO Algorithm	Increase convergence	Increase execution time
[5]	LFMVO Algorithm	The average best fit	MFO, MVO, PSO Algorithm	Increase the convergence speed compared to the compared algorithms	Early convergence
[30]	FPSO + FGA Algorithm	Optimization issues in 10 functions	PSO, GA Algorithm	Utilizing fuzzy logic to eliminate early convergence and entanglement in local optimization	Increase execution time
[26]	FFO Algorithm	One-objective optimization problems	-	Simple, easy to implement Group collaboration	Early Convergence and entanglement in local optimality

species, the fruit fly has stronger senses and perception, especially in vision and smell. The smell organs of fruit flies can find all kinds of scents suspended in the air, even when the food is located 40 km away. When a fruit fly gets close to the food location, it finds food using its vision and then flies in that direction. The mathematical model for the description of this algorithm is as follows [29]:

Step 1. The initial location of the swarm of fruit fly is randomly initialized.

$$\begin{aligned} &Init\ X_{axis} \\ &Init\ Y_{axis} \end{aligned} \tag{1}$$

Step 2. The random direction and distance to search of food for each fruit fly are defined.

$$\begin{aligned} X_i &= X_{axis} + Random\ Value \\ Y_i &= Y_{axis} + Random\ Value \end{aligned} \tag{2}$$

where $i = 1, 2, 3, ...$ population size of fruit flies.

Step 3. Because the food location cannot be known, first the distance to the origin is calculated (Dist), then the smell concentration judgment value (S) (the inverse of distance) is calculated.

$$Dist_i = \sqrt{X_i^2 + Y_i^2} \tag{3}$$

$$S_i = \frac{1}{Dist_i} \tag{4}$$

Step 4. The smell concentration judgment value of each fruit fly is replaced in the fitness function, and then the intensity of the smell of that location is found ($Smell_i$).

$$Smell_i = Function(S_i) \tag{5}$$

Step 5. The one fruit fly with minimum smell concentration among the fruit fly swarm is determined.

$$[bestSmell\ bestIndex] = min(Smell) \tag{6}$$

Step 6. If the intensity of the smell in this iteration is better than the previous value, then the best smell concentration value and (X, Y) coordinate are kept; at this moment, the fruit fly swarm can fly toward the food location using vision (Vision searching process).

$$\begin{aligned} &Smellbest = bestSmell \\ &X_{axis} = X(bestIndex) \\ &Y_{axis} = Y(bestIndex) \end{aligned} \tag{7}$$

Step 7. . Steps $2-6$ of the optimization algorithm are repeated until the stop condition is reached.

Algorithm 1. Pseudo code of Standard FAO Algorithm

for $gen = 1$ to gen_{max} **do**
  for $M = 1$ to $M = M_{pop}$ **do**
    Random initial the position of fruit fly population.
    $X_{\text{axis}} = rand()$
    $Y_{\text{axis}} = rand()$.
    Give fruit fly individuals a random distance and direction to randomly search for food using their sense of smell.
    $X = X_{axis} + 2 * rand() - 1$
    $Y = Y_{axis} + 2 * rand() - 1$
    Calculate the distance from the fruit fly to the origin (3)
    Compute the taste concentration smell using (4).
    Evaluate the taste concentration (Smell_i) using (5).
    Record bestSmell and bestIndex using (6).
    Update X_{axis} and Y_{axis}
  end for
  if $Smell_i > bestSmell$ **then**
    $X_{\text{axis}} = X(bestIndex)$
    $Y_{\text{axis}} = Y(bestIndex)$
  end if
end for

4. **Modified Fruit Fly Optimization (MFFOV)**

This paper presents a modified fruit fly optimization (MFFOV) with the aim of eliminating the local optimum and improving the accuracy and convergence speed of the fruit fly optimization algorithm. (Herein, we have n-number of fruit flies represented by $X = (x_1, x_2, ..., x_n)$ in D-dimensional, such that, i_{th} fruit fly, $x_i = (x_{i1}, x_{i2}, ..., x_{in})$).

4.1. **Adding velocity variable.** In the PSO algorithm, the particles are gathered to the most optimal point in each iteration; this operation is done intelligently (Figure 1). In the FFO algorithm, however, in each iteration, particles are distributed from the most optimal point without any intelligence (Figure 2). In this section, the next position of the particles will be made intelligently; thus, the change of speed and position is as follows:

$$\begin{aligned} V_{ix} &= w^* V_{ix} + c_1 rand()(P_{ix} - X_i) + c_2 rand()(G_x - X_i) \\ V_{iy} &= w^* V_{iy} + c_1 rand()(P_{iy} - Y_i) + c_2 rand()(G_y - Y_i) \end{aligned} \tag{8}$$

and

$$\begin{aligned} X_i(t+1) &= V_{ix}(t+1) + X_i(t) \\ Y_i(t+1) &= V_{iy}(t+1) + y_i(t) \end{aligned} \tag{9}$$

where w is inertia weight, c_1 and c_1 are acceleration constants, and rand() is random function. is the best local position experienced by $i - th$ fruit fly on the x direction, and P_{iy} is the best local position experienced by $i - th$ fruit fly on the y direction. G_x is the best position experienced by all fruit flies on the x direction, and G_y is the best position experienced by all fruit flies on the y direction.

4.2. **Random perturbation.** When the fitness value (Smellbest) compared with the fitness value in the previous iteration was not better, a random value (in search space) is added to swarm position. To find the best swarm location, this process can be put on a loop with a low number of iterations (we used $n = 10$).

$if\ bestSmell > Smellbest$

$X_{axis} = X(bestindex);$

$Y_{axis} = Y(bestindex);$

$Smellbest = bestSmell;$

$Else$

$For\ i = 1 : n$

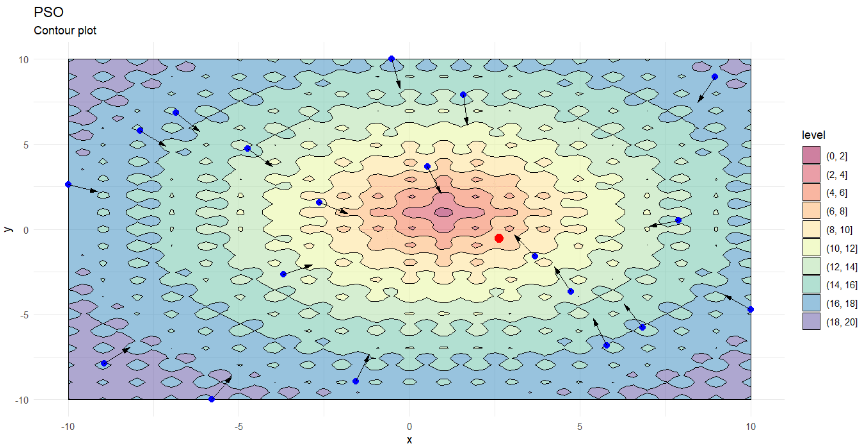

Figure 1. Update the position of the particles to the most optimal point (red point) in the sphere function in the PSO algorithm.

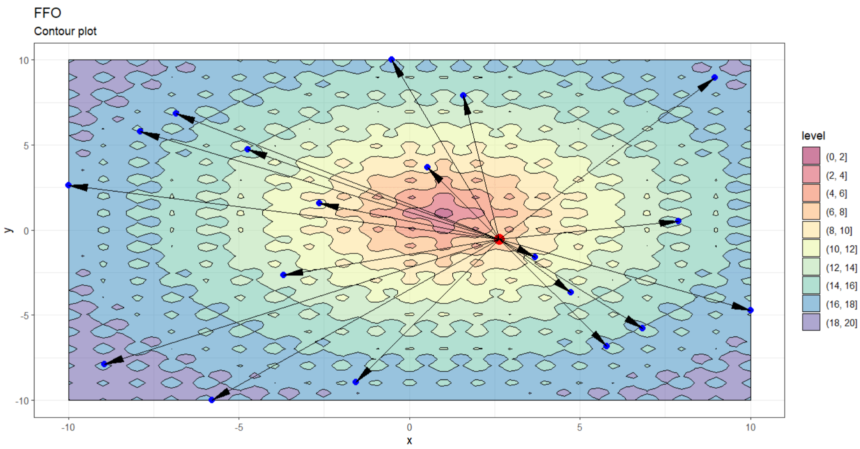

Figure 2. Update the position of the particles from the most optimal point (red point) in the sphere function in the FFO algorithm.

$$X_{new} = X(bestindex) + \frac{1}{2\pi} e^{\dfrac{-X(bestindex)^2}{2}}$$

$$Y_{new} = Y(bestindex) + \frac{1}{2\pi} e^{\dfrac{-Y(bestindex)^2}{2}}$$

$$Dist_{new} = \sqrt{(X_{new})^2 + (Y_{new})^2}$$

$$S_{new} = \frac{1}{Dist_{new}}$$

$$Smell_{new} = smellfunction(S_{new})$$

$if\ Smell_{new} > bestSmell$

$\quad Smellbest = Smell_{new}$

$\quad X_{axis} = X(new)$

$\quad Y_{axis} = Y(new)$

End

End

4.3. **Improve Smellbest based on dimension.** We can improve Smellbest by checking each dimension in the fruit fly position, such that if by substituting the value of one dimension (at a time) of the position of each fruit fly into the corresponding dimension value of Gbestposition, the fitness value improves, the Smellbest is updated.

(10) $For\, each\, dimension\, j,\ 1\, to\, D$

$For\, each\, fruit\, fly\, i,\ 1\, to\, n$

$Pos = Gbestposition$

$Pos = X_{ij}$

$if\ f(Pos) > Smellbest$

$Smellbest = f(Pos)$

$Gbestposition = Pos$

End

End

End

Gbestposition is the position of the best fruit fly in the current iteration. Smellbest is the smell of the best fruit fly in the current iteration. is -th dimension in the position of -th fruit fly.

Algorithm 2 Pseudo code of MFFOV Algorithm

for $gen = 1$ to gen_{max} **do**
    for $M = 1$ to $M = M_{pop}$ **do**
        Random initial the position of fruit fly population.
        $X_{axis} = rand()$
        $Y_{axis} = rand()$
        Guide the fruit fly individuals to the food according to the knowledge gained from the previous flies (8).
        $X = X_{axis} + V_{ix}$
        $Y = Y_{axis} + V_{iy}$
        Calculate the distance from the fruit fly to the origin (3)
        Compute the taste concentration smell using (4).
        Evaluate the taste concentration ($Smell_i$) using (5).
        Record bestSmell and bestIndex using (6).
        Update X_{axis} and Y_{axis}
    end for
    if $Smell_i > bestSmell$ **then**
        $X_{axis} = X(bestIndex)$
        $Y_{axis} = Y(bestIndex)$
    else
        Add a random value to the positions of the fruit fly individuals.
    end if
    for each dimension
        improve Smellbest based on Dimension (10).
    end for
end for

5. Results and Discussion

In this section, the MFFOV algorithm is evaluated on 10 test functions and compared to advanced swarm based optimization algorithms. Benchmark functions can be divided into two groups: unimodal and multi-modal function. These 10 benchmark functions are the classical functions used by many researchers [23]. From these 10 classical functions, the first eight are unimodal, and the second two are multi-modal. The unimodal functions (f1–f8) are suitable for determining algorithms' exploitation, because they have a global optimum and no local optimum. Multi-modal functions (f9, f10) have many local optima and are useful for examining the exploration and avoiding the algorithms' local optima. These benchmark functions are given in Table 4, where DIM represents the dimensions of the function and RANGE is the boundary of the functions. In this section, the following 8 metaheuristics are applied to the previously discussed problem: FFO, PSO, ABC, TLBO, GA, GSA, DE, and HPO. Table 2 compares the mean, variance, and mean computation time of at least 30 independent simulations between PSO, GA, FFO, and MFFOV.

To provide a fair comparison between the proposed algorithm and another selected set of algorithms, we followed the same initialization process for all the compared algorithms. For all experiments, the common parameter settings for all algorithms were as follows: The number of individuals used in the search process is set to 80, and the number of iterations is set to 1000. The values

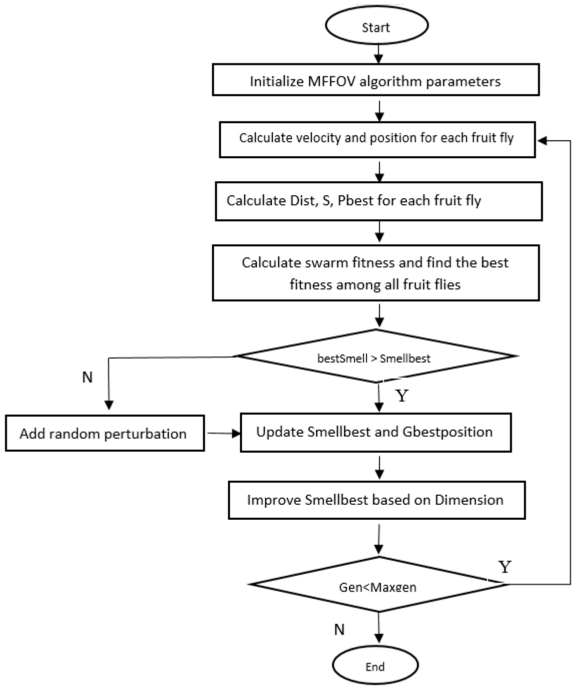

Figure 3. The flowchart of MFFOV.

Table 2. Comparison between PSO, GA, FFO and MFFOV for different typical fitness functions (the mean of calculation time).

Function	Dimension	Algorithm	Optimal value	Average value	Standard deviation	Running time/s
f_1	30	PSO	$4.11e-08$	$6.25e-07$	$7.11e-06$	5.697
	30	GA	0.1758	$9.75e-01$	$4.33e-01$	4.014
	30	FFO	0.0745	$1.85e+00$	$3.35e+00$	2.035
	30	MFFOV	0	0	0	3.865
f_5	30	PSO	0.0385	$1.85e+01$	$3.35e+01$	4.481
	30	GA	$4.96e-03$	$1.35e-02$	$7.28e-01$	3.239
	30	FFO	0.0985	$1.85e+00$	$3.35e+00$	1.967
	30	MFFOV	$3.13e-06$	$8.62e-05$	$4.18e-05$	3.175
f_9	30	PSO	$2.83e+03$	$2.15e+04$	$1.35e+01$	4.865
	30	GA	$1.43e+02$	$3.24e+03$	$6.01e+02$	3.451
	30	FFO	$5.47e+03$	$1.85e+04$	$3.35e+04$	12.074
	30	MFFOV	0	0	0	3.362

of c_1, c_2, will influence MFFOV performance. Here, we suggest and take the values 0.1 and 3.0, respectively. Furthermore, the value of w is 0.76 in the first iteration and w*0.99 in all subsequent iterations (Other parameters of each algorithm are shown in Table 3).

Table 3. The parameter settings.

Algorithm	Parameter	Value	Ref(s)
PSO	c_1 c_2	1.4961	[1]
	w	0.9	
TLBO	Teaching factor T	1, 2	[3]
GA	P_c	0.95	[1]
	P_m	0.001	
GSA	G_c	1	[22]
DE	CR	0.9	[1]
	F	0.95	
HPO	C	$\in [1, 0.002]$	[23]
	β	0.1	

Table 4. Benchmark functions used in experiments.

Function	D	Range	class	f_{min}
$f_1(x) = \sum_{i=1}^{D} x_i^2$	30	$[-100, 100]$	Unimodal	0
$f_2(x) = \mid x_i \mid + \prod_{i=1}^{D} \mid x_i \mid$	30	$[-10, 10]$	Unimodal	0
$f_3(x) = \sum_{i=1}^{D} (\sum_{j=1}^{i} x_j)^2$	30	$[-100, 100]$	Unimodal	0
$f_4(x) = max \mid x_i \mid, 1 \leqslant i \leqslant D$	30	$[-100, 100]$	Unimodal	0
$f_5(x) = \sum_{i=1}^{D} i x_i^4 + random[0, 1)$	30	$[-128, 128]$	Unimodal	0
$f_6(x) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	$[-30, 30]$	Unimodal	0
$f_7(x) = \sum_{i=1}^{D} ([x_i + 0.5])^2$	30	$[-100, 100]$	Unimodal	0
$f_8(x) = \sum_{i=1}^{D-1} -x_i \sin(\sqrt{\mid x_i \mid})$	30	$[-500, 500]$	multimodal	$-418.9829*5$
$f_9(x) = \frac{1}{4000} \sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} \cos(\frac{x_i}{\sqrt{i}}) + 1$	30	$[-600, 600]$	multimodal	0
$f_{10}(x) = 0.1\{\sin^2(3\pi x_1) + \sum_{i=1}^{D-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + (x_n - 1)[1 + \sin^2(2\pi x_n)]\} + \sum_{i=1}^{D} \mu(x_i, 5, 100, 4)$	30	$[-50, 50]$	multimodal	0

Table 5 summarizes the mean (ave) and standard deviation (stdv). Table 6 summarizes the Wilcoxon test p-values at the 0.05 significance level for MFFOV against the other seven algorithms. According to Table 5, MFFOV performed better than FFO, PSO, ABC, TLBO, GA, GSA, DE, and HPO in 1, 2, 3, 6, 9, and 10 benchmark test functions, respectively.

TABLE 5. Test Results of each Algorithm for each Function.

F	result	FFO	PSO	ABC	TLBO	GA	GSA	DE	HPO	MFFOV
f_1	avg	$1.85e+00$	$6.25e-07$	$1.45e-23$	$1.00e-152$	$9.75e-01$	$4.00e-57$	$8.57e-46$	0	0
	stdv	$3.35e+00$	$7.11e-06$	$6.99e-22$	$2.02e-152$	$4.33e-01$	$4.27e-55$	$2.38e-46$	0	0
f_2	avg	$1.85e+01$	$8.35e-04$	$8.35e-21$	$8.25e-26$	$4.72e-02$	$5.72e-32$	$2.55e-25$	$2.91e-92$	0
	stdv	$3.35e+01$	$6.26e-03$	$3.47e-18$	$8.51e-26$	$7.88e-01$	$4.32e-30$	$3.57e-25$	$1.11e-91$	0
f_3	avg	$1.85e+01$	$7.35e-05$	$1.25e-13$	$9.86e-21$	$5.44e-01$	$7.44e-65$	$9.39e-52$	$6.83e-144$	0
	stdv	$3.35e+01$	$3.44e-04$	$1.74e-13$	$1.82e-20$	$3.11e-01$	$6.24e-61$	$5.66e-52$	$3.72e-143$	0
f_4	avg	$1.85e+02$	$2.15e+00$	$4.36e-05$	$6.39e-19$	$4.75e+00$	$6.61e-43$	$8.25e-36$	$5.40e-77$	$4.22e-07$
	stdv	$3.35e+02$	$8.99e-01$	$7.78e-04$	$1.05e-54$	$1.42e+00$	$1.28e-43$	$3.76e-36$	$1.62e-76$	$5.62e-07$
f_5	avg	$1.85e+00$	$1.85e+01$	$5.91e-05$	$7.14e-00$	$1.35e-02$	$5.14e-08$	$5.14e-05$	$9.22e-37$	$8.62e-05$
	stdv	$3.35e+00$	$3.35e+01$	$4.41e-04$	$3.02e-00$	$7.28e-01$	$1.33e-08$	$1.74e-05$	$2.43e-35$	$4.18e-05$
f_6	avg	$1.85e+01$	$6.25e-07$	$5.65e-06$	$4.37e-13$	$2.38e-01$	$4.74e-25$	$5.57e-33$	0	0
	stdv	$3.35e+01$	$2.12e-06$	$1.58e-06$	$1.22e-13$	$2.48e-01$	$4.39e-24$	$3.67e-33$	0	0
f_7	avg	$1.85e+02$	$2.41e-01$	$5.34e-11$	$4.34e-18$	$2.77e-03$	$9.45e-29$	0	0	$4.62e-29$
	stdv	$3.35e+02$	$6.63e-02$	$1.11e-09$	$7.52e-16$	$4.62e-02$	$4.77e-29$	0	0	$5.55e-29$
f_8	avg	$1.85e+05$	$-6.54e+04$	$-4.10e+04$	$-7.14e+02$	$-6.67e+05$	$-1.22e+01$	$-5.35e+02$	$-6.24+02$	$-2.85e+02$
	stdv	$3.35e+05$	$3.73e+02$	$2.10e+02$	$1.03e+02$	$7.84e+05$	$2.04e+01$	$07.11e+02$	$5.88+02$	$4.11e+02$
f_9	avg	$1.85e+03$	$2.15e+04$	$1.54e+04$	0	$3.24e+04$	0	0	0	0
	stdv	$3.35e+03$	$1.35e+01$	$5.44e+03$	0	$6.01e+03$	0	0	0	0
f_{10}	avg	$1.85e+02$	$8.24e+04$	$3.14e+04$	$2.20e-03$	$2.17e+04$	$7.76e-09$	$2.54e-08$	0	0
	stdv	$3.35e+02$	$1.41e+02$	$7.92e+02$	$4.44e-03$	$8.14e+03$	$9.32e-09$	$4.62e-08$	0	0

Table 6. P-values of the Wilcoxon test.

Function	FFO	PSO	ABC	TLBO	GA	GSA	DE	HPO	MFFOV
	P-value	P-value	P-value	P-value	P-value	P-value	P-value	P-value	P-value
f_1	$1.78e-06$	$1.78e-06$	$1.78e-06$	$1.78e-06$	$1.78e-06$	$1.78e-06$	$1.78e-06$	NAN	NAN
f_2	$1.78e-06$	$1.78e-06$	$1.78e-06$	$1.77e-06$	$1.78e-06$	$1.78e-06$	$1.78e-06$	$1.78e-06$	NAN
f_3	$1.78e-06$	$1.78e-06$	$1.78e-06$	$1.78e-06$	$1.78e-06$	$1.78e-06$	$1.78e-06$	$1.78e-06$	NAN
f_4	$1.77e-06$	$1.77e-06$	$1.76e-06$	$1.77e-06$	$1.78e-06$	$1.78e-06$	$1.77e-06$	$1.78e-06$	$1.78e-06$
f_5	$1.78e-06$	$1.78e-06$	$1.78e-06$	$1.78e-06$	$1.78e-06$	$1.78e-06$	$1.78e-06$	$1.78e-06$	$1.78e-06$
f_6	$1.78e-06$	$1.74e-06$	$1.78e-06$	$1.78e-06$	$1.76e-06$	$1.78e-06$	$1.78e-06$	NAN	NAN
f_7	$1.78e-06$	$1.78e-06$	$1.78e-06$	$1.78e-06$	$1.78e-06$	$1.78e-06$	NAN	NAN	$1.78e-06$
f_8	$1.78e-06$	$1.78e-06$	$1.78e-06$	$1.78e-06$	$1.78e-06$	$1.78e-06$	$1.78e-06$	$1.78e-06$	$1.78e-06$
f_9	$1.77e-06$	$1.78e-06$	$1.78e-06$	NAN	$1.78e-06$	NAN	NAN	NAN	NAN
f_{10}	$1.78e-06$	$4.65e-04$	$2.52e-05$	$1.78e-06$	$1.78e-06$	$3.78e-03$	$1.76e-06$	NAN	NAN

5.1. **Qualitative Analysis of MFFOV.** Figure 4 presents the qualitative results, including search landscapes, convergence curves, average fitness curves in logarithmic shapes, search history and trajectory of the first individual, associated with the MFFOV algorithm in solving a selected set of test functions for up to 200 iterations.

6. Performance of MFFOV Algorithm on Constrained Problems

MFFOV was also tested with two constrained engineering design problems: a tension/compression spring and a three-bar truss. Unlike basic test functions, real-world problems have equality and inequality constraints; therefore, MFFOV should be equipped with a constraints control method to optimize such problems. The performance of the algorithm in dealing with constrained optimization problems is significantly influenced by the employed constraint handling technique (CHT). In recent decades, many constraint control methods have been developed for optimization algorithms. Some popular CHTs among them are the death penalty, co-evolutionary, adaptive, annealing, dynamic and static [21]. The death penalty function, the simplest method, assigns a big objective value. It eliminates impossible solutions by optimization algorithms during the optimization process. The advantages of this method are

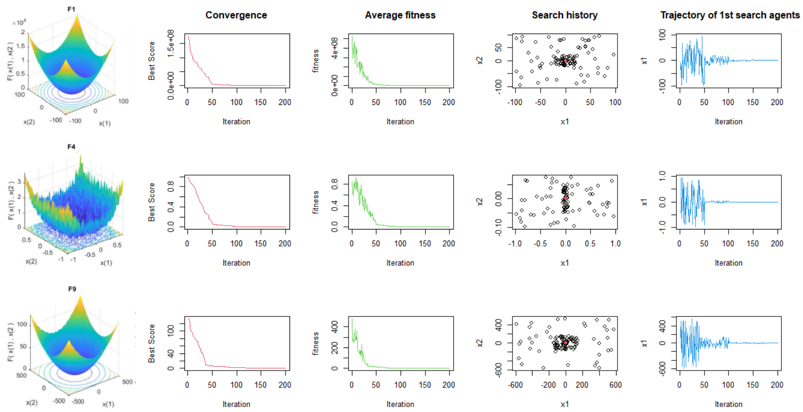

FIGURE 4. Convergence behavior and search history of the proposed MFFOV algorithm.

low computational costs and simplicity [11]. The methods were compared using the death penalty method, because most of the algorithms used the same. The results of the MFFOV algorithm were compared with the algorithms that previously solved these problems. The number of search agents was set to 80, and the maximum number of iterations was set to 1000.

6.1. **Three-bar truss design problem.** Consider the three-bar truss design shown in figure 5, taken from Naruei [23]. This problem involved two variables and three inequality constraints. The design optimization problem can be formulated as follows:

$$
\begin{aligned}
min \quad & f(X) = (2\sqrt{2}x_1 + x_2) * l \\
s.t. \quad & g_1(X) = \frac{\sqrt{2}x_1 + x_1}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \leq 0 \\
& g_2(X) = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \leq 0 \\
& g_2(X) = \frac{1}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \leq 0 \\
& Where\, 0 \leq x_1 \leq 1,\, 0 \leq x_2 \leq 1,\, l = 100cm,\, P = 2\frac{KN}{cm^2},\, and\, \sigma = 2\frac{KN}{cm^2}.
\end{aligned}
$$

The proposed MFFOV method was applied to the three-bar truss design problem, and the optimal solutions were compared to earlier results reported by CS [11], WCA [10], MFO [19], and OBSCA [1]. The MFFOV algorithm achieved better results than all algorithms except WCA (Table 7).

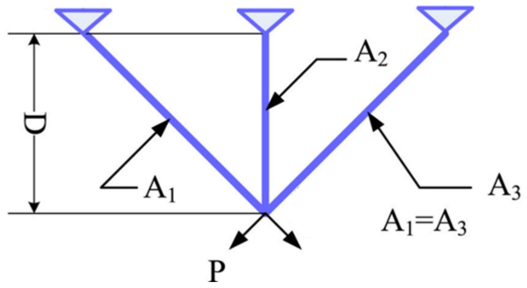

FIGURE 5. Three-bar truss design problem [23].

TABLE 7. Comparison results for three-bar truss design problem.

Algorithm	Optimal values for variables		constraints			Optimal weight
	X_1	X_2	g_1	g_2	g_3	
CS [11]	0.78867	0.40902	−0.00029	−0.26853	−0.73176	263.9716
WCA [10]	0.788651	0.408316	−0.000000	−1.464024	−0.535975	263.8958
MFO [19]	0.78901	0.44025	−0.0241817	−1.440996	−0.5831856	267.1907
OBSCA [1]	0.77457	0.46647	−0.01173657	−1.406186	−0.6055509	265.7285
MFFOV	0.78847	0.408902	−0.0000054	−1.463386	−0.536669	263.9032

6.2. **Tension/compression spring.** The tension/compression spring design problem (Figure 6.) is described in [19]. The design optimization problem involves three continuous variables and four nonlinear inequality constraints.

$$
\begin{aligned}
min \quad & f(X) = (x_3+2)x_2x_1^2 \\
s.t. \quad & g_1(X) = 1 - \frac{x_2^3x_3}{71785x_1^4} \leq 0 \\
& g_2(X) = \frac{1}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \leq 0 \\
& g_3(X) = 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0 \\
& g_4(X) = \frac{x_1+x_2}{1.5} - 1 \leq 0 \\
Where\, & 0.25 \leq x_1 \leq 1.3,\ 0.05 \leq x_2 \leq 2,\ 2 \leq x_3 \leq 15
\end{aligned}
$$

The proposed MFFOV method was applied to the tension/compression spring design problem, and the optimal solutions were compared to earlier results reported by MPM [3], WCA [10], PSO and MVO [22]. The MFFOV algorithm achieved better results than all other algorithms (Table 8).

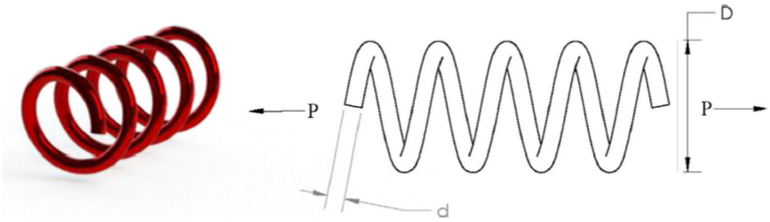

FIGURE 6. Tension/compression spring design problem [23].

TABLE 8. Comparison results for r tension/compression spring.

Algorithm	Optimal values for variables			constraints				Optimal weight
	$X_1(d)$	$X_2(D)$	$X_3(N)$	$g_1(x)$	$g_2(x)$	$g_3(x)$	$g_4(x)$	
MPM [3]	0.05000	0.31590	14.2500	-0.000014	-0.003782	-3.938302	-0.756067	0.0128334
WCA [10]	0.051689	0.356718	11.288957	$-1.65e-13$	$-7.90e-14$	-4.053399	-0.727864	0.012665
PSO	0.05000	0.310414	15.0000	$-3.30e-06$	-0.01737	-3.858675	-0.759724	0.0131926
MVO [22]	0.05000	0.315956	14.22623	-0.0001287	-0.0036433	-3.9448	-0.756029	0.01281694
MFFOV	0.051691	0.35677	11.285441	$-2.28e-05$	$-1.26e-05$	-4.054073	-0.727692	0.01266468

7. Conclusions

Targeting the phenomenon of easily relapsing into local extremum and low convergence accuracy of the fruit fly optimization algorithm, this paper proposes an adaptive fruit fly optimization algorithm based on velocity in particle swarm optimization, improved Smellbest based on dimension, and random perturbation. Experimental results for a set of benchmark test functions and engineering design problems seem to show that hybridization provides a more effective trade-off between exploitation and exploration of the search space. The proposed MFFOV algorithm has been shown to be competitive and efficient when compared with the others; however, further research is required to examine the efficiencies of the proposed algorithm on other real-world optimization and large-scale optimization problems.

References

[1] Abd Elaziz, M., Oliva, D., & Xiong, S. (2017). An improved opposition-based sine cosine algorithm for global optimization. Expert Systems with Applications, 90, 484-500. https://doi.org/10.1016/j.eswa.2017.07.043

[2] Abdullahi, M., & Ngadi, M. A. (2016). Symbiotic organism search optimization based task scheduling in cloud computing environment. Future Generation Computer Systems, 56, 640-650. https://doi.org/10.1016/j.future.2015.08.006

[3] Belegundu, A. D., & Arora, J. S. (1985). A study of mathematical programming methods for structural optimization. Part I: Theory. International Journal for Numerical Methods in Engineering, 21(9), 1583-1599. https://doi.org/10.1002/nme.1620210904

[4] Bezdan, T., Stoean, C., Naamany, A. A., Bacanin, N., Rashid, T. A., Zivkovic, M., & Venkatachalam, K. (2021). Hybrid fruit-fly optimization algorithm with k-means for text document clustering. Mathematics, 9(16), 1929. https://doi.org/10.3390/math9161929

[5] Bouchekara, H. R. E. H., Zellagui, M., & Abido, M. A. (2017). Optimal coordination of directional overcurrent relays using a modified electromagnetic field optimization algorithm. Applied Soft Computing, 54, 267-283. https://doi.org/10.1016/j.asoc.2017.01.037

[6] Chen, P. W., Lin, W. Y., Huang, T. H., & Pan, W. T. (2013). Using fruit fly optimization algorithm optimized grey model neural network to perform satisfaction analysis for e-business service. Applied Mathematics & Information Sciences, 7(2L), 459-465. http://dx.doi.org/10.12785/amis/072L12

[7] Darvish, A., & Ebrahimzadeh, A. (2018). Improved fruit-fly optimization algorithm and its applications in antenna arrays synthesis. IEEE transactions on antennas and propagation, 66(4), 1756-1766. https://doi.org/10.1109/TAP.2018.2800695

[8] Ding, G., Dong, F., & Zou, H. (2019). Fruit fly optimization algorithm based on a hybrid adaptive-cooperative learning and its application in multilevel image thresholding. Applied Soft Computing, 84, 105704. https://doi.org/10.1016/j.asoc.2019.105704

[9] Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization. IEEE computational intelligence magazine, 1(4), 28-39. https://doi.org/10.1109/MCI.2006.329691

[10] Eskandar, H., Sadollah, A., Bahreininejad, A., & Hamdi, M. (2012). Water cycle algorithm–A novel metaheuristic optimization method for solving constrained engineering optimization problems. Computers & Structures, 110, 151-166. https://doi.org/10.1016/j.compstruc.2012.07.010

[11] Gandomi, A. H., Yang, X. S., & Alavi, A. H. (2013). Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Engineering with computers, 29, 17-35. https://doi.org/10.1007/s00366-011-0241-y

[12] Hashim, F. A., & Hussien, A. G. (2022). Snake Optimizer: A novel meta-heuristic optimization algorithm. Knowledge-Based Systems, 242, 108320. https://doi.org/10.1016/j.knosys.2022.108320

[13] Hussain, K., Mohd Salleh, M. N., Cheng, S., & Shi, Y. (2019). Metaheuristic research: a comprehensive survey. Artificial intelligence review, 52, 2191-2233. https://doi.org/10.1007/s10462-017-9605-z

[14] Iscan, H., & Gunduz, M. (2017). An application of fruit fly optimization algorithm for traveling salesman problem. Procedia computer science, 111, 58-63. https://doi.org/10.1016/j.procs.2017.06.010

[14] Karaboga, D. (2010). Artificial bee colony algorithm. scholarpedia, 5(3), 6915. http://dx.doi.org/10.4249/scholarpedia.6915

[15] Kaveh, A., & Bakhshpoori, T. (2016). Water evaporation optimization: a novel physically inspired optimization algorithm. Computers & Structures, 167, 69-85. https://doi.org/10.1016/j.compstruc.2016.01.008

[16] Kennedy, J., & Eberhart, R. (1995, November). Particle swarm optimization. In Proceedings of ICNN'95-international conference on neural networks (Vol. 4, pp. 1942-1948). IEEE. https://doi.org/10.1109/ICNN.1995.488968

[17] Li, H. Z., Guo, S., Li, C. J., & Sun, J. Q. (2013). A hybrid annual power load forecasting model based on generalized regression neural network with fruit fly optimization algorithm. Knowledge-Based Systems, 37, 378-387. https://doi.org/10.1016/j.knosys.2012.08.015

[18] Li, Y., & Han, M. (2020). Improved fruit fly algorithm on structural optimization. Brain informatics, 7, 1-13. https://doi.org/10.1016/j.knosys.2012.08.015

[19] Mirjalili, S. (2015). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. Knowledge-based systems, 89, 228-249. https://doi.org/10.1016/j.knosys.2015.07.006

[20] Mirjalili, S. (2016). SCA: a sine cosine algorithm for solving optimization problems. Knowledge-based systems, 96, 120-133. https://doi.org/10.1016/j.knosys.2015.12.022

[21] Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. Advances in engineering software, 95, 51-67. https://doi.org/10.1016/j.advengsoft.2016.01.008

[22] Mirjalili, S., Mirjalili, S. M., & Hatamlou, A. (2016). Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Computing and Applications, 27, 495-513. http://dx.doi.org/10.1007/s00521-015-1870-7

[23] Naruei, I., Keynia, F., & Sabbagh Molahosseini, A. (2022). Hunter–prey optimization: Algorithm and applications. Soft Computing, 26(3), 1279-1314. https://doi.org/10.1007/s00500-021-06401-0

[24] Noroozi, M., Mohammadi, H., Efatinasab, E., Lashgari, A., Eslami, M., & Khan, B. (2022). Golden search optimization algorithm. IEEE Access, 10, 37515-37532. https://doi.org/10.1007/s10462-017-9605-z

[25] Pan, Q. K., Sang, H. Y., Duan, J. H., & Gao, L. (2014). An improved fruit fly optimization algorithm for continuous function optimization problems. Knowledge-Based Systems, 62, 69-83. https://doi.org/10.1016/j.knosys.2014.02.021

[26] Pan, W. T. (2012). A new fruit fly optimization algorithm: taking the financial distress model as an example. Knowledge-Based Systems, 26, 69-74. https://doi.org/10.1016/j.knosys.2011.07.001

[27] Sayed, G. I., Darwish, A., & Hassanien, A. E. (2018). A new chaotic multi-verse optimization algorithm for solving engineering optimization problems. Journal of Experimental & Theoretical Artificial Intelligence, 30(2), 293-317. https://doi.org/10.1080/0952813X.2018.1430858

[28] Swami, V., Kumar, S., & Jain, S. (2018). An improved spider monkey optimization algorithm. In Soft Computing: Theories and Applications: Proceedings of SoCTA 2016, Volume 1 (pp. 73-81). Springer Singapore. https://doi.org/10.1007/978-981-10-5687-17

[29] Torabi, S., & Safi-Esfahani, F. (2018). A dynamic task scheduling framework based on chicken swarm and improved raven roosting optimization methods in cloud computing. The Journal of Supercomputing, 74(6), 2581-2626. https://doi.org/10.1007/s11227-018-2291-z

[30] Valdez, F., Melin, P., & Castillo, O. (2011). An improved evolutionary method with fuzzy logic for combining particle swarm optimization and genetic algorithms. Applied Soft Computing, 11(2), 2625-2632. https://doi.org/10.1016/j.asoc.2010.10.010

[31] Wang, L., Zheng, X. L., & Wang, S. Y. (2013). A novel binary fruit fly optimization algorithm for solving the multidimensional knapsack problem. Knowledge-Based Systems, 48, 17-23. https://doi.org/10.1016/j.knosys.2013.04.003

[32] Wu, L., Liu, Q., Tian, X., Zhang, J., & Xiao, W. (2018). A new improved fruit fly optimization algorithm IAFOA and its application to solve engineering optimization problems. Knowledge-Based Systems, 144, 153-173. https://doi.org/10.1016/j.knosys.2017.12.031

[33] Yazdani, M., & Jolai, F. (2016). Lion optimization algorithm (LOA): a nature-inspired metaheuristic algorithm. Journal of computational design and engineering, 3(1), 24-36. https://doi.org/10.1016/j.jcde.2015.06.003

[34] Zhang, X., Xu, Y., Yu, C., Heidari, A. A., Li, S., Chen, H., & Li, C. (2020). Gaussian mutational chaotic fruit fly-built optimization and feature selection. Expert Systems with Applications, 141, 112976. https://doi.org/10.1016/j.eswa.2019.112976

[35] Zhao, F., Qin, S., Zhang, Y., Ma, W., Zhang, C., & Song, H. (2019). A two-stage differential biogeography-based optimization algorithm and its performance analysis. Expert Systems with Applications, 115, 329-345. https://doi.org/10.1016/j.eswa.2018.08.012

Reza Etesami
Orcid number: 0000-0003-4141-8852
Department of Statistics
Faculty of Mathematics & Computer
Shahid Bahonar University of Kerman
Kerman, Iran
Email address: rezaetesamii@math.uk.ac.ir

Mohsen Madadi
Orcid number: 0000-0002-7950-138X
Department of Statistics
Faculty of Mathematics & Computer
Shahid Bahonar University of Kerman
Kerman, Iran
Email address: madadi@uk.ac.ir

Farshid Keynia
Orcid number: 0000-0002-9027-7315
Department of Energy Management and Optimization
Institute of Science and High Technology and Environmental Sciences
Graduate University of Advanced Technology
Kerman, Iran
Email address: f.keynia@kgut.ac.ir