Solving Fractional Programming by Improving Firefly Algorithm

Juncheng Guo, Shouchuan Liu, Yonghong Zhang, and Zhijian Duan

Abstract—**Engineering and economics both make extensive use of fractional programming. Because they are highly nonconvex and multimodal, they are regarded as challenging. This paper proposes an enhanced firefly algorithm (HFA) for solving fractional programming . The new population mean center is predicted by using the historical data of the population mean centers and added to the movement equation of fireflies to better guide their search. Numerical experiments are provided to demonstrate the efficiency and robustness of HFA. The results obtained by HFA show that it is always better than those produced by other methods.**

Index Terms—**Firefly algorithm, Optimization, Swarm intelligence, Mean-based prediction**

I. INTRODUCTION

THE following fractional programming is considered in this paper:

$$\text{FP}\begin{cases} \min & f(x) = \sum_{i=1}^{p} \gamma_i \dfrac{g_i(x)}{h_i(x)} \\ \text{s.t.} & Ax \le b, \\ & X^0 = \{x \in R^n \mid l^0 \le x \le u^0\} \end{cases}$$

wher $p \geq 2$, $g_i(x) = \sum_{j=1}^{n} c_{ij} x_j + d_i, h_i(x) = \sum_{j=1}^{n} e_{ij} x_j + f_i$ are finite affine functions such that $g_i(x) > 0$, $h_i(x) > 0$ for all $x \in \Lambda \triangleq \{x \mid Ax \le b, x \in X0\}$, $A = (a_{ij})_{m \times n}$, $b \in R^m$, and γ_i are real constant coefficients, $i = 1, \cdots, p$.

Currently, FP is one of the most successful fields in nonlinear optimization. Up to now, extensive research on specific cases of FP has been carried out. The first reason is that FP has many important applications in many domains, such as multiobjective bond portfolio [1], cluster analysis [2], government contracting challenges [3], etc. The second reason is that it usually poses significant theoretical and computational issues since it has multiple local optima that are not globally optima. Furthermore, the goal function is NP-hard and neither quasiconvex nor quasiconcave [4].

FP has so far been addressed by a variety of techniques. A parametric simplex strategy, for example, has been devised [5] for $p = 2$. Alternatively, numerous algorithms have been presented [6,7] for $p \geq 2$ that iteratively search the nonconvex outcome space until a globally optimal solution is identified. Additionally, a number of different branch and bound based methods have been suggested for specific FP cases [8-14].

Although deterministic algorithms for global optimal solutions of fractional programming have made great progress, most of these methods are limited to FP without coefficients γ_i. A more general model is considered in this study.

The aim of this paper is to develop an enhanced firefly algorithm (HFA) to solve FP. The main features of this algorithm are as follows: (1) Only special cases of the problem (FP) may be handled by the previously studied approaches (e.g., [9,11,12,14]); in contrast, the suggested algorithm HFA can solve the general FP. (2) Compared to Shen and Wang's approach, HFA does not require to introduce extra variables [13]. (3) The HFA's movement equation looks for a potential area by incorporating the population mean's historical data. HFA can handle almost all the test problems in finding globally optimal solutions, as shown in numerical experiment.

This paper is organized as follows. The basic FA is introduced in Section II. Section III provides a description of the suggested algorithm HFA. Section IV presents some numerical results, and Section V offers some conclusions.

II. BASIC FA ALGORITHM

Scholars have shown a great deal of interest in the firefly algorithm (FA) since it was proposed by Yang in 2008 [15]. The job shop scheduling problem [16], vision-based railway overhead inspection system [17], visual tracking [18], network reconfiguration of unbalanced distribution networks [19], and many other problems were solved by FA.

FA imitates the flashing behavior of fireflies. Every firefly in the search space is viewed as a potential solution, and their movement patterns show how solutions are upgraded in quest

Manuscript received March 17, 2024; revised August 8, 2024.

This work was supported in part by the Scientific and Technological Research Projects in Henan Province(242102210146); the Key Scientific Research Projects in Universities in Henan Province (24B110006); the Research Project of Henan Polytechnic(2024J046); the Basic Research Program of Natural Science of Shaanxi Province (2024JC-YBMS-003, 2024JC-YBMS-013); the Key Cultivation Project of Xianyang Normal University (XSYK21044); the Research Project on Teaching Reform of Xianyang Normal University (2023C117); the 14th Five-Year Plan for Education Science in Shaanxi Province (SGH23Y2506).

Juncheng Guo is an associate professor of the Basic Education Department, Henan Polytechnic, Zhengzhou, 450046, PR China. (email:acheng2090@126.com).

Shouchuan Liu is a lecturer of the Henan Geology Mineral College, Zhengzhou, 451464, PR China. (e-mail: 286538517@qq.com).

Yonghong Zhang is an associate professor of the School of Mathematics and Statistics, Xianyang Normal University, Xianyang, 712000, PR China. (Corresponding author e-mail: zhangyonghong09@126.com).

Zhijian Duan is an associate professor of the School of Mathematics and Statistics, Xianyang Normal University, Xianyang, 712000, PR China. (e-mail: email:zhijian_duan@126.com).

of better ones.

In addition, three idealized hypotheses are tested to make sure the algorithm is headed in the right direction. They are shown as follows:

(1) Fireflies are asexual, which means that every firefly will be attracted to brighter fireflies without being affected by gender.

(2) The brightness of the flashing is correlated with the attraction's size. If x_i is more depressed than x_j for two random fireflies, x_i is drawn to x_j. In the event that x_i is the best firefly available, it will travel randomly through the search range.

(3) The firefly's brightness varies according to the objective function's value. Author List

A. Basic parameters

Assume that the search space has dimension D and that the swarm size is ps.

First, in the search space, the initial population in FA is generated randomly as follows:

$$x_i = l + rand \cdot (u - l),\ i = 1, \cdots, ps, \tag{1}$$

where l and u are the lower and upper bounds of the search space.

In FA, both attractiveness (β) and brightness (I) are important variables. Generally speaking, there is a negative correlation for minimum difficulties and a positive correlation for maximum problems between the firefly's brightness and the objective function value.

The following calculation of the Euclidean distance r_{ij} between x_i and x_j is required before determining the attractiveness β:

$$r_{ij} = \sqrt{\sum_{t=1}^{D} (x_{it} - x_{jt})^2}. \tag{2}$$

Then, attractiveness β is calculated using the formula below:

$$\beta = \beta_0 \cdot e^{-\gamma \times r_{ij}^2}, \tag{3}$$

in which γ represents the absorption coefficient and β_0 is the maximum attractiveness.

B. The movement equation of the firefly

Assuming that x_i is drawn to x_j, x_i will travel according to the formula below:

$$x_i^{t+1} = x_i^t + \beta \cdot (x_j^t - x_i^t) + \alpha \cdot r_1, \tag{4}$$

where the position of x_i at the t-th iteration is represented by x_i^t, $r_1 \in$ [-0.5, 0.5], and $\alpha \in$ [0, 1] is a random step.

C. The basic steps of FA

Step 1: Set parameters α, β_0, population size ps, and then initialize the population.

Step 2: Use formulas (2) and (3) to calculate r_{ij} and β, respectively.

Step 3: Use formula (4) to update the solution.

Step 4: Compare the present solution with the prior solution to get a superior one.

Step 5: Determine the objective function value for the new solution.

Step 6: If the algorithm satisfies the terminate condition, it terminates and produces the best result; if not, move back to Step 2.

Since FA was proposed, many variations have been developed, which fall into three categories: (1) The algorithm can quickly enter local optima due to the fixed step α. To address this limitation, variable strategies for step setting were used to adapt FA [20-23]. (2) To adopt the advantages of other swarm intelligence algorithms, some hybrid algorithms have been proposed. In order to address the combinatorial optimization of non-slicing VLSI floor planning, Sivaranjani and Kumar [24] devised the hybrid particle swarm optimization-firefly method (HPSOFA). In FA, the brightest firefly moves at random, but the others, aside from the darker one, hardly move at all. Three innovative operators were used in a hybrid optimizer by Wang et al. [25]. The pattern search algorithm was used in [26] as a local optimization method to improve the firefly algorithm. Wang et al. [27] combined FA with differential evolution (DE) to improve the search for the brighter one. Rahmani and Mirhassani [28] suggested a hybrid firefly-Genetic Algorithm to address the capacitated facility locating problem. Goel and Maini [29] integrated the ant colony and FA to address vehicle routing difficulties and enhance FA's performance. By using the visual function of the similarity of fireflies and butterflies, Zhang et al. [30] presented a novel hybrid FA with butterfly optimization algorithm (BOA), namely FA-BOA. To solve job shop scheduling problems, Nugraheni et al. [31] designed a hybrid metaheuristics based on genetic algorithm (GA) and FA. (3) To enhance FA's performance even more, some scholars blend it with traditional optimization methods. Gandomi et al. [32], for instance, integrated chaos with FA to enhance the solution's resilience and exploration. In order to improve FA, Kotteeswaran and Sivakumar [33] created Lévy -flight.

In basic FA, all fireflies that are brighter than a particular darker firefly can attract it. As a result, it may increase the computational complexity of time and easily lead to oscillations during the search. Wang et al.[34] presented a novel FA in 2016 called the firefly algorithm with random attraction (RaFA) in an effort to lessen its detrimental consequences. In their method, not all fireflies, but only the firefly x_j, which is randomly selected among the rest aside from x_i, can attract firefly x_i. As a result, the time complexity is decreased significantly. But, it cannot ensure that x_i should be directed in a better way; instead, it can slow down the algorithm's accuracy and convergence. Wang et al.[35] presented an enhanced FA (NaFA) in order to get around this drawback. In NaFA, x_i found a k-neighbor surrounding it based on a circular topology. If x_j, a member of the k-neighbor, is brighter than x_i, then x_i will travel in the direction of x_j.

To further enhance the performance of FA, this paper tries to incorporate the past population's collective experience into the movement equation. Thus, each firefly can be guided by this integration to look for potentially fruitful areas.

III. The improved FA:HFA

In FA, the brighter fireflies have the ability to attract each firefly several times. Nonetheless, not all of population evolution's experience has been put to good use. To fully use the historical experience, HFA will predict the population mean center and introduce it into the movement equation to better direct population dynamics. Here is the detailed procedure.

Let $Mean(0)$ be the mean center of the initial population. $\alpha = 0.1$ is a preset value, which will determine the extent to which the historical mean influences the prediction results. Let $Mean(t)$ be the mean center of the population at the t-th iteration, $YC(t+1)$ be the prediction value about $Mean(t+1)$. $YC(t+1)$ is defined as follows:

$$YC(t+1) = Mean(t) + \phi \cdot YC(t) \quad (5)$$

where $YC(0) = Mean(0)$, $\phi = 0.1$.

Obviously, $YC(t+1)$ incorporates the mean information of the previous populations, and the influence of the early population mean information weakens as the iteration progresses.

Based on $YC(t+1)$, we propose a novel movement equation for fireflies, which is given as follows:

$$x_i^{t+1} = x_i^t + \beta_1 \cdot (x_j^t - x_i^t) + \beta_2 \cdot r_1 \cdot (YC(t+1) - x_i^t) + \alpha \cdot r_2,$$

where α is a random step, $r_1 \in [-1,1], r_2 \in [-0.5, 0.5]$, and x_i^t denotes the position of x_i at the t-th iteration.

The algorithm steps of HFA are as follows:

Step 1: Set parameters α, β_0, the population size ps, and initialize the population.

Step 2: Use formulas (2) and (3) to calculate r_{ij} and β, respectively.

Step 3: Use formulas (5) and (6) to update the solution.

Step 4: Compare the present solution with the prior solution to get a superior one.

Step 5: Determine the objective function value for the new solution.

Step 6: If the algorithm satisfies the terminate condition, it terminates and produces the best result; if not, move back to Step 2.

IV. Numerical results and analysis

Two experiments are conducted in order to evaluate HFA's performance. 14 widely used benchmark functions that were taken from CEC 2005 are chosen for Experiment 1 and evaluated with FA, RaFA, and NaFA. In Experiment 2, the effectiveness of HFA in solving fractional problems is evaluated by contrasting it with a number of conventional techniques, such as those found in [10], [13], [14], and [36]. These experiments are carried out on a Windows 7 PC using Matlab 2017a (Win 64) with an Inter(R) Core(TM) i5-4258U CPU running at 2.40 GHz.

A. Experiment 1: test on benchmark functions

Table I provides the details of these functions. The dimensions, search space bounds, and global minimum values of these functions are denoted by D, Range, and $f(x^*)$, respectively. In this experiment, the population size is 50, and each algorithm runs independently on each test function 30 times to ensure fair comparison. 3000 is the maximum number of iterations that can be done, and it also serves as the termination condition. The proposed comparison techniques are applied to the remaining parameters. Minimum value (Min), mean value (Mean) and standard deviation (Std) are used to compare their performance. Table II provides a summary of the comparison results.

TABLE I
Benchmark test functions

Function	$f_1 = \sum_{i=1}^{D} x_i^2$
Range	[-100,100]
D	30
Optimal Value	0
Function	$f_2 = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} \lvert x_i \rvert$
Range	[-10,10]
D	30
Optimal Value	0
Function	$f_3 = \sum_{i=1}^{D} (\sum_{j=1}^{i} x_j)^2$
Range	[-100,100]
D	30
Optimal Value	0
Function	$f_4 = \max\{\lvert x_i \rvert, 1 \le i \le D\}$
Range	[-100,100]
D	30
Optimal Value	0
Function	$f_5 = \sum_{i=1}^{D} i x_i^2$
Range	[-100,100]
D	30
Optimal Value	0
Function	$f_6 = \sum_{i=1}^{D} i x_i^4$
Range	[-100,100]
D	30
Optimal Value	0
Function	$f_7 = \sum_{i=1}^{D} \lvert x_i \rvert^{(i+1)}$
Range	[-100,100]
D	30
Optimal Value	0
Function	$f_8 = \sum_{i=1}^{D} (10^6)^{\frac{i-1}{D-1}} x_i^2$
Range	[-100,100]
D	30
Optimal Value	0

Function	$f_9=\sum_{i=1}^{D}(\lfloor x_i+0.5\rfloor)^2$
Range	[-100,100]
D	30
Optimal Value	0
Function	$f_{10}=\sum_{i=1}^{D} i x_i^4+random[0,1)$
Range	[-100,100]
D	30
Optimal Value	0
Function	$f_{11}=\sum_{i=1}^{D}(x_i^2-10\cos(2\pi x_i)+10)$
Range	[-100,100]
D	30
Optimal Value	0
Function	$f_{12}=-20\exp(-0.2\sqrt{\sum_{i=1}^{D} x_i^2/D}) - \exp(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i))+20+e$
Range	[-100,100]
D	30
Optimal Value	0
Function	$f_{13}=0.5+\frac{\sin(\sqrt{\sum_{i-1}^{D} x_i^2})^2-0.5}{(1+0.001\sum_{i-1}^{D} x_i^2)^2}$
Range	[-100,100]
D	30
Optimal Value	0
Function	$f_{14}=\sum_{i=1}^{D}\lvert x_i sin(x_i)+0.1x_i\rvert$
Range	[-100,100]
D	30
Optimal Value	0

From Table II, we can see that HFA can find their optimal solutions on functions f_1, f_2, f_5-f_9, f_{11}, and f_{14}, accounting for 57% of the test problem. For f_9, although FA can also find its optimal solution, its robustness is poor and the mean deviation is large. For f_3 and f_4, the accuracy of HFA is much higher than that of other algorithms. The comparison results show that the performance of HFA is far superior to other algorithms.

To intuitively compare the convergence rate of HFA and the other three FAs, the convergence curves of these algorithms are displayed on f_9, f_{10}, f_{11}, f_{13} and f_{14} in Figs 1-5. From Figs 1-5, it is easy to see that HFA can find a better solution when the algorithm terminates. Especially for f_9, HFA converges to the optimal solution at a faster speed.

The comparison results indicate that the mean prediction information introduced by HFA is of great useful in improving its performance.

TABLE II
THE COMPARISON RESULTS FOR DIFFERENT ALGORITHMS

Function		f_1
FA	Min	0
	Mean	0
	SD	0
NaFA	Min	2.154e+03
	Mean	2.706e+03
	SD	7.797e+02
RaFA	Min	7.409e+02
	Mean	9.717e+02
	SD	3.236e+02
HFA	Min	0
	Mean	0
	SD	0
Function		f_2
FA	Min	4.249e-322
	Mean	4.693e-322
	SD	0
NaFA	Min	3.155e+00
	Mean	5.197e+00
	SD	1.483e+00
RaFA	Min	6.603e+00
	Mean	7.845e+00
	SD	1.566e+00
HFA	Min	0
	Mean	0
	SD	0
Function		f_3
FA	Min	1.166e+03
	Mean	1.841e+03
	SD	8.842e+02
NaFA	Min	1.764e+03
	Mean	2.900e+03
	SD	1.410e+03
RaFA	Min	1.531e+03
	Mean	2.483e+03
	SD	1.021e+02
HFA	Min	5.060e-273
	Mean	7.667e-267
	SD	0
Function		f_4
FA	Min	4.392e-14
	Mean	2.019e-02
	SD	1.947e-02
NaFA	Min	2.541e+01
	Mean	2.637e+01
	SD	1.307e+00
RaFA	Min	1.168e+01
	Mean	1.223e+01
	SD	7.869e-01
HFA	Min	2.002e-242
	Mean	4.239e-240
	SD	0
Function		f_5
FA	Min	4.608e-04
	Mean	3.904e+00
	SD	1.455e+00

NaFA	Min	2.384e+02
	Mean	2.546e+02
	SD	2.303e+01
RaFA	Min	1.918e+02
	Mean	2.153e+02
	SD	2.767e+01
HFA	Min	0
	Mean	0
	SD	0
Function		f_6
FA	Min	0
	Mean	0
	SD	0
NaFA	Min	1.507e-01
	Mean	5.884e-01
	SD	6.190e-01
RaFA	Min	6.302e-02
	Mean	1.218e-01
	SD	7.672e-02
HFA	Min	0
	Mean	0
	SD	0
Function		f_7
FA	Min	4.230e-07
	Mean	4.351e-07
	SD	1.707e-08
NaFA	Min	7.919e-05
	Mean	5.146e-04
	SD	6.158e-04
RaFA	Min	1.593e-07
	Mean	5.097e-06
	SD	6.984e-06
HFA	Min	0
	Mean	0
	SD	0
Function		f_8
FA	Min	1.482e+06
	Mean	7.078e+06
	SD	5.302e+02
NaFA	Min	2.530e+07
	Mean	3.044e+07
	SD	1.605e+03
RaFA	Min	9.086e+06
	Mean	1.281e+07
	SD	5.273e+06
FA	Min	0
	Mean	0
	SD	0
Function		f_9
FA	Min	0
	Mean	5.103e-01
	SD	7.071e-01
NaFA	Min	1.0e+00
	Mean	1.0e+00
	SD	0
RaFA	Min	3.0e+00
	Mean	3.501e+00
	SD	1.224e-01
HFA	Min	0
	Mean	0
	SD	0
Function		f_{10}
FA	Min	2.320e-02
	Mean	3.191e-02
	SD	1.023e-02
NaFA	Min	2.678e-01
	Mean	3.517e-01
	SD	1.185e-01
RaFA	Min	6.601e-02
	Mean	8.192e-02
	SD	2.263e-02
HFA	Min	5.775e-06
	Mean	6.045e-06
	SD	3.812e-07
Function		f_{11}
FA	Min	5.285e+01
	Mean	5.621e+01
	SD	1.634e+00
NaFA	Min	4.137e+01
	Mean	6.623e+01
	SD	2.267e+01
RaFA	Min	3.365e+01
	Mean	4.898e+01
	SD	1.199e+01
HFA	Min	0
	Mean	0
	SD	0
Function		f_{12}
FA	Min	1.332e-14
	Mean	1.758e-14
	SD	3.891e-15
NaFA	Min	3.667e-01
	Mean	4.321e-01
	SD	6.981e-02
RaFA	Min	6.106e-01
	Mean	7.721e-01
	SD	1.439e-01
HFA	Min	9.770e-15
	Mean	1.758e-14
	SD	3.891e-15
Function		f_{13}
FA	Min	1.270e-01
	Mean	2.850e-01
	SD	1.297e-01
NaFA	Min	4.999e-01
	Mean	5.012e-01
	SD	2.656e-05
RaFA	Min	1.298e-01
	Mean	1.789e-01
	SD	1.850e-02
HFA	Min	9.700e-02
	Mean	9.700e-02
	SD	2.060e-13
Function		f_{14}
FA	Min	7.581e-02
	Mean	1.683e-01
	SD	6.992e-02
NaFA	Min	1.009e+01

	Mean	1.137e+01
	SD	1.049e+00
RaFA	Min	7.761e+00
	Mean	1.068e+01
	SD	1.812e+00
HFA	Min	0
	Mean	0
	SD	0

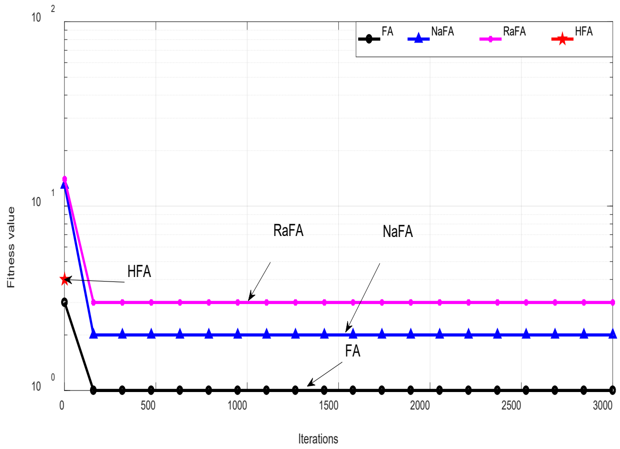

Fig 1 The convergence curves of different algorithms on f_9

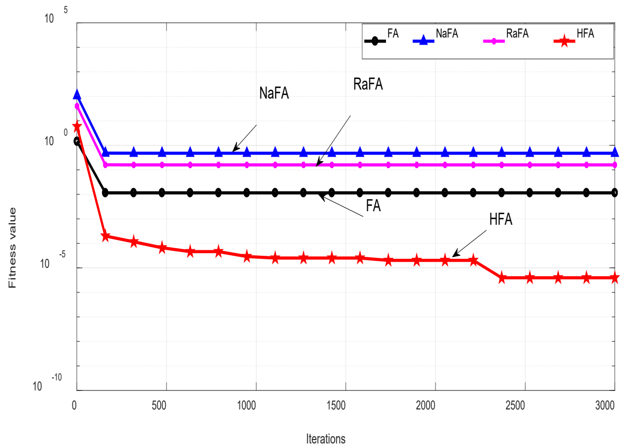

Fig 2 The convergence curves of different algorithms on f_{10}

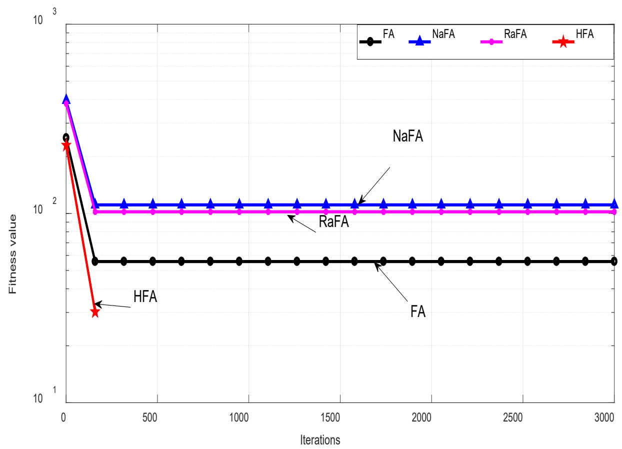

Fig 3 The convergence curves of different algorithms on f_{11}

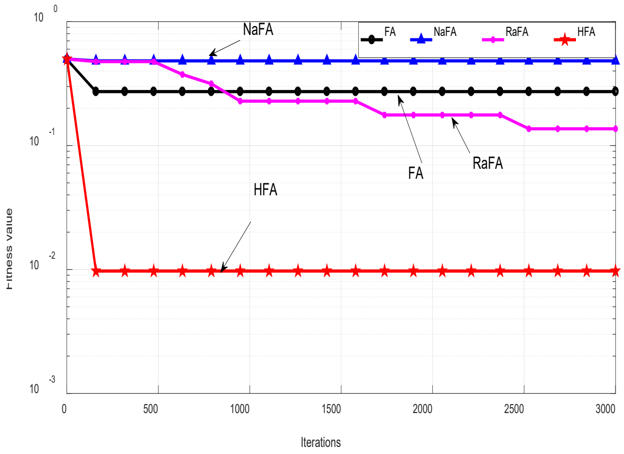

Fig 4 The convergence curves of different algorithms on f_{13}

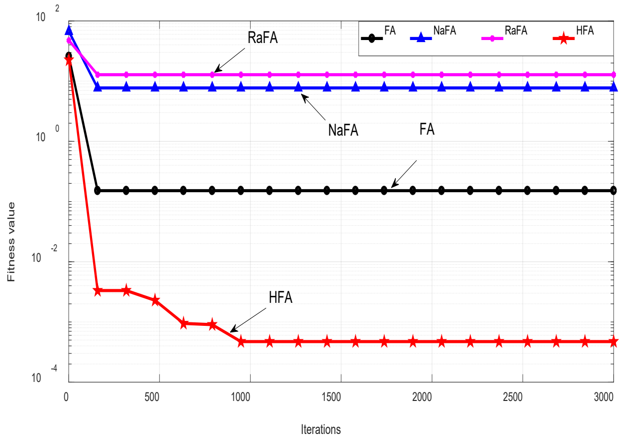

Fig 5 The convergence curves of different algorithms on f_{14}

B. *Experiment 2: test on fractional problems*

In this subsection, HFA is used to solve several fractional problems, and is compared with some methods in [10], [13], [14], and [36]. The convergence tolerance is set to ϵ=1.0e−8, just like what is given in the comparative literatures.

Example 1. (Shen and Wang [13])

$$\begin{aligned} \min \quad & -\frac{3x_1+4x_2+50}{3x_1+5x_2+4x_3+50}+\frac{3x_1+5x_2+3x_3+50}{5x_1+5x_2+4x_3+50} \\ & +\frac{x_1+2x_2+4x_3+50}{5x_2+4x_3+50}+\frac{4x_1+3x_2+3x_3+50}{3x_2+3x_3+50} \\ \text{s.t.} \quad & 6x_1+3x_2+3x_3 \le 10, \\ & 10x_1+3x_2+8x_3 \le 10, \\ & x_1 \ge 0, x_2 \ge 0, x_3 \ge 0. \end{aligned}$$

Example 2. (Wang et al. [14])

$$\begin{aligned} \min \quad & -\frac{4x_1+3x_2+3x_3+50}{3x_2+3x_3+50}-\frac{3x_1+4x_3+50}{4x_1+4x_2+5x_3+50} \\ & -\frac{x_1+2x_2+5x_3+50}{x_1+5x_2+5x_3+50}-\frac{x_1+2x_2+4x_3+50}{5x_2+4x_3+50} \\ \text{s.t.} \quad & 2x_1+x_2+5x_3 \le 10, \\ & x_1+6x_2+3x_3 \le 10, \\ & 5x_1+9x_2+2x_3 \le 10, \\ & 9x_1+7x_2+3x_3 \le 10, \\ & x_1 \ge 0, x_2 \ge 0, x_3 \ge 0. \end{aligned}$$

Example 3. (Jiao [36])

$$\min \ -\frac{3x_1+5x_2+3x_3+50}{3x_1+4x_2+5x_3+50}-\frac{3x_1+4x_2+50}{4x_1+3x_2+2x_3+50} -\frac{4x_1+2x_2+4x_3+50}{5x_1+4x_2+3x_3+50}$$

$$\text{s.t.} \quad 6x_1+3x_2+3x_3 \le 10, \quad 10x_1+3x_2+8x_3 \le 10, \quad 0 \le x_1 \le 1, \ 0 \le x_2 \le 3.3333, \ 0 \le x_3 \le 1.$$

Example 4. (Shen and Wang [13])

$$\min \ -\frac{37x_1+73x_2+13}{13x_1+13x_2+13}+\frac{63x_1-18x_2+39}{13x_1+26x_2+13} -\frac{13x_1+13x_2+13}{63x_2-18x_3+39}+\frac{13x_1+26x_2+13}{37x_1+73x_2+13}$$

$$\text{s.t.} \quad 5x_1-3x_2=3, \quad 1.5 \le x_1 \le 3.$$

Example 5.(Jiao [36])

$$\min \ \frac{-x_1+2x_2+2}{3x_1-4x_2+5}+\frac{4x_1-3x_2+4}{-2x_1+x_2+3}$$

$$\text{s.t.} \quad x_1+x_2 \le 1.5, \quad x_1-x_2 \le 0, \quad 0 \le x_1 \le 1, \ 0 \le x_2 \le 1.$$

Example 6.(Ji et al. [10])

$$\min \ \frac{x_1+3x_2+2}{4x_1+x_2+3}+\frac{4x_1+3x_2+1}{x_1+x_2+4}$$

$$\text{s.t.} \quad -x_1-x_2 \le -1, \quad 0 \le x_1, \ 0 \le x_2.$$

The comparison results are given in Table III. Among these comparison results, except for HFA, all other results are directly from the corresponding comparative literature.

From Table III, it can be seen that HFA can find better results than the comparative algorithms, at least not inferior to the their results. Moreover, HFA does not require branching of the search region and constructing linear relaxation functions. Therefore, HFA is an effective method for solving fractional programming problems.

TABLE III
COMPUTATIONAL RESULTS OF TEST EXAMPLES 1-6

Example	Method	Optimal Solution	Optimal Value
1	[13]	(0.0, 3.333333, 0.0)	1.9
	HFA	(0.0,3.3333333,0.0)	1.9
2	[14]	(1.0, 0.0, 0.0)	-4.081481
	HFA	(1.111111, 0.0, 0.0)	-4.0907029
3	[15]	(0.0, 3.3333, 0.0)	-3.0029239
	HFA	(0.0, 3.333333, 0.0)	-3.0029239
4	[13]	(3.0, 4.0)	-3.29167
	HFA	(3.0, 4.0)	-3.2916666
5	[15]	(0.0, 0.283935)	1.6231833
	HFA	(0.0, 0.2839473925)	1.6231833
6	[10]	(1.0, 0.0)	1.42857
	HFA	(1.0, 0.0)	1.4285714

V. CONCLUSION

In this work, we integrated empirical knowledge into the FA algorithm to direct fireflies toward qualified region, hence improving individual guidance for movement. The efficiency of this mechanism was shown by the numerical experiments. We intend to use this approach in the future to solve some real-world issues.

REFERENCES

[1] H. Konno, and H. Watanabe, “Bond portfolio optimization problems and their applications to index tracking:A partial optimization approach,” *Journal of the Operations Research Society of Japan*, vol. 39, no. 3, pp. 295-306, 1996.

[2] M.R. Rao, “Cluster analysis and mathematical programming,” *Journal American Statistical Association*, vol. 66, no. 335, pp. 622-626, 1971.

[3] C.S. Colantoni, R.P. Manes, and A. Whinston, “Programming, profit rates, and pricing decisions,” *Accounting Review,* vol. 44, no. 3, pp. 467-481, 1969.

[4] T. Matsui, “NP-hardness of linear multiplicative programming and related problems,” *Journal of Global Optimization*, vol. 9, no. 2, pp. 113-119, 1996.

[5] H. Konno, Y. Yajima, and T. Matsui, “Parametric simplex algorithms for solving a special class of nonconvex minimization problem,” *Journal of Global Optimization,* vol.1, no. 1, pp. 65-81, 1991.

[6] J.E. Falk, and S.W. Palocsay, “Optimizing the sum of Linear fractional functions,” Recent Advance in Global Optimization, Edited by C.A. Floudas and P.M. Pardalos, Princeton University press, Princeton, New Jersey, 1992.

[7] J.E. Falk, and S.W. Palocsay, “Image space analysis of generalized fractional programs,” *Journal of Global Optimization,* vol. 4, no. 1, pp. 63-88, 1994.

[8] H.P. Benson,“On the global optimization of sums of linear fractional functions over a convex set,” *Journal of Optimization Theory and Applications,* vol. 121, no. 1, pp. 19-39, 2004.

[9] H.P. Benson, “A simplicial branch and bound duality-bounds algorithm for the linear sum-of-ratios problem,” *European Journal of Operational Research,* vol. 182, no. 2, pp. 597-611, 2007.

[10] Y. Ji, K.C. Zhang, and S.J. Qu, “A deterministic global optimization algorithm,” *Applied Mathematics and Computation,* vol. 185, no. 1, pp. 382-387, 2007.

[11] H. Konno, N. Abe, “Minimization of the sum of three linear fractional functions,” *Journal of Global Optimization,* vol. 15, no. 4, pp. 419-432, 1999.

[12] H. Konno, and K. Fukaish, “A branch and bound algorithm for solving low rank linear multiplicative and fractional programming problems,” *Journal of Global Optimization,* vol. 18, no. 3, pp. 283-299, 2000.

[13] P.P. Shen, and C.F. Wang, “Global optimization for sum of linear ratios problem with coefficients,” *Journal of Global Optimization,* vol. 176, no. 1, pp. 219-229, 2006.

[14] Y.J. Wang, P.P. Shen, and Z.A. Liang, “A branch-and-bound algorithm to globally solve the sum of several linear ratios,” *Applied Mathematics and Computation*, vol. 168, no. 1, pp. 89-101, 2005.

[15] X.S. Yang, " Nature-Inspired Metaheurstic Algorithms," Luniver Press, 2008.

[16] K.C. Udaiyakumar, and M. Chandrasekaran, “Application of firefly algoritnm in job shop scheduling problem,” *Microcomputer Applications*, vol. 97, pp. 1798-1807, 2014.

[17] I. Aydin, “A new approach based on firefly algorithm for vision-based railing overhead inspection system," *Measurement,* vol. 74, pp. 43-55, 2015.

[18] M.L. Gao, and L.L. Li et al., “Firefly algorithm (FA) based particle filter method for visual tracking,” *Optik-International Journal for Light and Electron Optics,* vol. 126, no. 18, pp. 1705-1711, 2015.

[19] M. Kaur, and S. Ghosh, “Network reconfiguration of unbalanced distribution networks using fuzzy-firefly algorithm,” *Applied Soft Computing,* vol. 49, pp. 868-886, 2016.

[20] S. Yu, and S. Zhu et al., “A variable step size firefly algorithm for numerical optimization,” *Applied Mathematics and Computation,* vol. 263, pp. 214-220, 2015.

[21] S. Yu, and S. Su et al., “A novel vise step strategy for firefly algorithm,” *International Journal of Computation Mathematics,* vol. 91, no. 12, pp. 2507-2513, 2014.

[22] I. Fister, and X.S. Yang et al., “Memetic firefly algorithm for combinatorial optimization,” *in Proceeding of the 5th International Conference on Bioinspired Optimization Methods and their Applications,* 2012.

[23] H. Wang, and X. Zhou et al., “Firefly algorithm with adaptive control parameters,” *Soft Computing,* vol. 21, no. 17, pp. 5091-5102, 2017.

[24] P. Sivaranjani, and A. S. Kumar, “Hybrid particle swarm optimization-firefly algorithm (HPSOFA) for combinatorial

optimization of non-slicing VLSI floorplanning," *Journal of Intelligent and Fuzzy Systems,* vol. 32, no. 1, pp. 661-669, 2017.

[25] X.W. Xia, and L. Gui et al., "A hybrid optimizer based on firefly algorithm and particle swarm optimization algorithm," *Journal of Computational Science,* vol. 26, pp. 488-500, 2018.

[26] A.M. Beigi, and A. Maroosi, "Parameter identification for solar cells and module using a hybrid firefly and pattern search algorithms," *Solar Energy,* vol. 171, no. 1, pp. 435-446, 2018.

[27] W. Wang, and H. Wang et al., "A hybrid firefly algorithm for continuous optimizatin problems," *International Conference on Cloud Computing and Security,* vol. 10040, pp. 522-531, 2016.

[28] A. Rahmani, and S.A. Mirhassani, "A hybrid firefly-genetic algoritnm for the capacitated facility location problem," *Information Sciences,* vol. 283, pp. 70-78, 2014.

[29] R. Goel, and R. Maini, "A hybrid of ant colony and firefly algorithms (HAFA) for solving vehicle routing problems," *Journal of Computational Science,* vol. 25, pp. 28-37, 2018.

[30] J.Q. Zhang, X. F. Xie, M. Wang, and M.J. Zhang, "A hybrid firefly algorithm with butterfly optimization algorithm and its Application," *Engineering Letters,* vol. 30, no. 2, pp. 453-462, 2022.

[31] C.E. Nugraheni, D. Swastiani, and L. Abednego, "Hybrid metaheuristics for job shop scheduling problems," *Engineering Letters,* vol. 30, no. 4, pp. 1444-1451, 2022.

[32] A. H. Gandomi, and X. S. Yang et al., "Firefly algorithm with chaos, " *Communications in Nonlinear Science and Numerical Simulation,* vol. 18, no. 1, pp. 89-98, 2013.

[33] R. Kotteeswaran, and L. Sirakumar, "Levy guided firefly algorithm based tuning of decentralised PI controller of nonlinear multivariable system-Coal Gasifier," *IFAC Proceedings,* vol. 47, no. 1, pp. 127-134, 2014.

[34] H. Wang, and W.J. Wang et al., "Firefly algorithm with random attraction," *International Journal of Bio-Inspired Computation,* vol. 8, no. 1, pp. 33-41, 2016.

[35] H. Wang, and W.J. Wang et al., "Firefly algorithm with neighborhood attraction," *Information Sciences,* vol. 382, pp. 374-387, 2017.

[36] H.W. Jiao, "A branch and bound algorithm for globally solving a class of nonconvex programming problems," *Nonlinear Analysis-Theory Methods & Applications,* vol. 70, no. 2, pp. 1113-1123, 2008.