International Journal of
Intelligent Systems and Applications in Engineering
ISSN:2147-6799 www.atscience.org/IJISAE
Advanced Technology and Science
Original Research Paper

A Modified Flower Pollination Algorithm forFractional Programming Problems

Mohamed Abdel-Baset*[1], Ibrahim M. Hezam[2]

Accepted 15th August 2015 DOI: 10.18201/ijisae.56680

Abstract:Flower pollination algorithm is a new nature-inspired algorithm, based on the characteristics of flowering plants. In this paper, a new method is developed chaos-based Flower Pollination Algorithm (CFPA) to solveFractional Programming Problems (FPPs). The proposed algorithm is tested using several ROP benchmarks. The test aims to prove the capability of the CFPA to solve any type of FPPs. The solution results employing the CFPA algorithm arecompared with a number of exact and metaheuristic solution methods used for handling FPPs. Numerical examples are given to show the feasibility, effectiveness, and robustness of the proposed algorithm. The results obtained using CFPAindicated the superiority of the proposed technique among others in computational time.

Keywords:*Flower Pollination Algorithm; Nature-Inspired Algorithm; Optimization; Chaos; Fractional Programming Problems.*

1. Introduction

The general from of the fractional programming problemsFPPs mathematical modelis as follows[1]

$$\min/\max \quad z(x_1,\dots,x_n)=\sum_{i=1}^{p}\frac{f_i(x)}{g_i(x)}$$
$$\textit{subject to} \quad x\in S, x\geq 0$$
$$S=\left\{x\in R^n \left| \begin{array}{l} h_k(x)\geq 0, \quad k=1,\dots,K; \\ m_j(x)=0, \quad j=1,\dots,J; \\ x_i^l\leq x_i\leq x_i^u, \; i=1,..,n. \end{array}\right.\right\} \tag{1}$$
$$g_i(x)\neq 0, \; i=1,..,p.$$
$$x_i\geq 0; \quad i=1,..,n.$$

where $f_i(x)$, $g_i(x)$, are supposed to be continuous functions, S is compact. Fractional programming problem sof the form (1) arise reality whenever rates such as the ratios (profit/revenue), (profit/time), (-waste of raw material/quantity of used raw material), are to be maximized often these problems are linear concave-convex fractional programming.

Fractional programmingproblems is a nonlinear programming method that has known increasing exposure recently and its importance, in solving concrete problems, is steadily increasing. Furthermore, nonlinear optimization models describe practical problems much better than the linear optimization, with many assumptions, does. The fractional programming problems are particularly useful in the solution of economic problems in which different activities use certain resources in different proportions. While the objective is to optimize a certain indicator, usually the most favorable return on allocation ratio subject to the constraint imposed on the availability of resources. It also has a number of important practical applications in manufacturing, administration, transportation, data mining, etc.

The solution methods to solve fractional programming problems can be broadly classified into classified into exact and inexact approaches. Some examples of traditional approaches is that of ([2] who introduced the parametric approach, [3]solved the linear fractional programming problems by converting Fractional ProgrammingProblems FPP into an equivalent linear programming problem and solved it using already existing standard algorithms for Linear Programming ProblemLPP, [4], [5] reviewed some of the methods that treated solving the FPP as the primal and dual simplex algorithm. The crisscross, which is based on pivoting, within an infinite number of iterations, either solves the problem or indicates that the problem is infeasible or unbounded. The interior point method, as well as Dinkelbach algorithms both reduces the solution of the Linear Fractional Programming LFP problem to the solution of a sequence of Linear Programming LP problems. Isbell Marlow method,Martos' Algorithm, Cambini–Martein's Algorithm, Bitran and Novae's Method, Swarup's Method. Moreover, there are many recent approaches employing traditional mathematical methods for solving the Fractional ProgrammingProblem FPPs such as: [6-7].

A few studies in recent years used heuristics approaches to solve fractional programmingproblems. [8] presented a genetic algorithm based method to solve the linear fractional programming problems. A set of solution point are generated using random numbers, feasibility of each solution point is verified, and the fitness value for all the feasible solution points are obtained. Among the feasible solution points, the best solution point is found out and then replaces the worst solution point. A pairwise crossover method is used for crossover and a new set of solution points is obtained. These steps are repeated

[1] *Department of Operations Research, faculty of Computers and Informatics, Zagazig University, El-**ZeraSquare**, **Zagazig**,Sharqiyah, Egypt*
[2] *Department of computer, Faculty of Education, Ibb University, Ibb city, Yemen.*
* *Corresponding Author:Email: analyst_mohamed@yahoo.com*

for a certain number of generations and the best solution for the given problem is obtained. [9] developed a genetic algorithm for the class of bi-level problems in which both level objective functions are linear fractional and the common constraint region is a bounded polyhedron. [1] proposed algorithm for the sum-of-ratios problems based harmony search algorithm. [10] developed neural networks for nonlinear fractional programming problem. The research proposed a new projection neural network model. It's theoretically guaranteed to solve variational inequality problems. The multiobjectiveminimax nonlinear fractional programming was defined and its optimality is derived by using its Lagrangian duality. The equilibrium points of the proposed neural network model are found to correspond to the Karush Kuhn Trcker point associated with the nonlinear fractional programming problem. [11] presented a neural network method for solving a class of linear fractional optimization problems with linear equality constraints. The proposed neural network model have the following two properties. First, it is demonstrated that the set of optima to the problems coincides with the set of equilibria of the neural network models which means the proposed model is complete. Second, it is also shown that the model globally converges to an exact optimal solution for any starting point from the feasible region. [12]Used particle swarm optimization algorithm for solving fractional programming problems.[13–16] introduced solution for integer fractional programming problem and complex variable fractional programming problems based swarm intelligence under uncertainty.

Flower pollination is an intriguing process in the natural world[17], [18]. Its evolutionary characteristics can be used to design new optimization algorithms. The algorithm obtained good results were dealing with lower-dimensional optimization problems[17], but may become problematic for higher-dimensional problems because of its tendency to converge very fast initially. This paper introduced an improved Flower pollination algorithm by integrating it with chaosto improve the reliability of the global optimality, also enhances the quality of the results.

This paper is organized as follows: after introduction, the original Flower pollination algorithm isbriefly introduced. In section 3, the proposed algorithm is described. section 4 introduces themeaning of chaos, while the results are discussed in section 5. Finally, conclusions are presented in section 6.

2. The Original Flower Pollination Algorithm

Flower Pollination Algorithm (FPA) was founded by Yang in the year 2012. Inspired by the flow pollination process of flowering plants are the following rules:

Rule 1: Biotic and cross-pollination can be considered as a process of global pollination process, and pollen-carrying pollinators move in a way that obeys Le'vy flights.

Rule 2: For local pollination, a biotic and self-pollination are used.

Rule 3: Pollinators such as insects can develop flower constancy, which is equivalent to a reproduction probability that is proportional to the similarity of two flowers involved.

Rule 4: The interaction or switching of local pollination and global pollination can be controlled by a switch probability $p \in [0,1]$, with a slight bias toward local pollination.

In order to formulate updating formulas, we have to convert the aforementioned rules into updating equations. For example, in the global pollination step, flower pollen gametes are carried by pollinators such as insects, and pollen can travel over a long distance because insects can often fly and move in a much longer range[17].Therefore, Rule 1 and flower constancy can be represented mathematically as:

$$x_i^{t+1} = x_i^t + \text{¥}L(\text{ª})(x_i^t - B) \tag{2}$$

Where x_i^t is the pollen i or solution vector $x_i$at iteration t, and $B$is the current best solution found among all solutions at the current generation/iteration. Here γ is a scaling factor to control the step size. In addition, $L(\lambda)$ is the parameter that corresponds to the strength of the pollination, which essentially is also the step size. Since insects may move over a long distance with various distance steps, we can use a Le'vy flight to imitate this characteristic efficiently. That is, we draw $L > 0$ from a Levy distribution:

$$L \sim \frac{\text{ª}\Gamma(\text{ª})\sin(\text{®ª}/2)}{\text{®}} \frac{1}{S^{1+\text{ª}}}, (S >> S_0 > 0) \tag{3}$$

Here, Γ(λ) is the standard gamma function, and this distribution is valid for large steps $s > 0$.

Then, to model the local pollination, both Rule 2 and Rule 3 can be represented as:

$$x_i^{t+1} = x_i^t + U(x_j^t - x_k^t) \tag{4}$$

Where x_j^t and x_k^t are pollen from different flowers of the same plant species. This essentially imitates the flower constancy in a limited neighborhood. Mathematically, if x_j^t and x_k^t comes from the same species or selected from the same population, this equivalently becomes a local random walk if we draw U from a uniform distribution in [0, 1].Though Flower pollination activities can occur at all scales, both local and global, adjacent flower patches or flowers in the not-so-far-away neighborhood are more likely to be pollinated by local flower pollen than those faraway. In order to imitate this, we can effectively use the switch probability like in Rule 4 or the proximity probability p to switch between common global pollination to intensive local pollination. To begin with, we can use a naive value of $p = 0.5$as an initially value. A preliminary parametric showed that $p = 0.8$ might work better for most applications[17].

The basic steps of FPA can be summarized as the pseudo-code shown in Figure 1.

```
Flower pollination algorithm[17]
Define Objective functionf (x), x = (x_1, x_2, ..., x_d)
Initialize a population of n flowers/pollen gametes with random solutions
Find the best solution Bin the initial population
Define a switch probability p    [0, 1]
while (t <MaxGeneration)
for i = 1 : n (all n flowers in the population)
if rand <p,
Draw a (d-dimensional) step vector L which obeys a Lévydistribution
Global pollination via $x_i^{t+1} = x_i^t + L(B - x_i^t)$
else
DrawUfrom a uniform distribution in [0,1]
Do local pollination via $x_i^{t+1} = x_i^t + U(x_j^t - x_k^t)$
end if
Evaluate new solutions
If new solutions are better, update them in the population
end for
Find the current best solution B
end while
```

```
Output the best solution found
```

Figure 1. Pseudo code of the Flower pollination algorithm

3. Chaos Theory

Chaos is a deterministic, random-like process found in nonlinear, dynamical system, which is
non-period, non-converging and bounded. Moreover, it depends on its initial condition and parameters[19-22]. Applications of chaos in several disciplines including operations research, physics, engineering, economics, biology, philosophy and computer science[23].
Recently chaos has been extended to various optimization areas because it can more easily escape from local minima and improve global convergence in comparison with other stochastic optimization algorithms [24]. Using chaotic sequences in flower pollination algorithm can be helpful to improve the reliability of the global optimality, and enhance the quality of the results**.**

3.1. Chaotic maps

At random-based optimization algorithms, the methods using chaotic variables instead of random variables are called chaotic optimization algorithms (COA) [20]. In these algorithms, due to the non-repetition and ergodicity of chaos, it can carry out overall searches at higher speeds than stochastic searches that depend on probabilities [25-26]. To achieve this issue, herein one-dimensional, noninvertible maps are utilized to generate chaotic sets. We will illustrate some of well-known one-dimensional maps as:

3.1.1. Logistic map

The Logistic map is defined by:

$$Y_{n+1} = \mu Y_n(1-Y_n) Y\in(0,1) 0 < \mu \le 4 \quad (5)$$

3.1.2. The Sine map

The Sine map is written as the following equation:

$$Y_{n+1} = \frac{\mu}{4}\sin(\pi Y_n)\, Y\epsilon(0,1)\ 0 < \mu \le 4 \quad (6)$$

3.1.3. Iterative chaotic map

The iterative chaotic map with infinite collapses is described as:

$$Y_{n+1} = \sin\left(\frac{\mu\pi}{Y_n}\right)\mu \in (0,1) \quad (7)$$

3.1.4. Circle map

The Circle map is expressed as:

$$Y_{n+1} = Y_n + \alpha - \left(\frac{\beta}{2\pi}\right)\sin(2\pi Y_n)\, mod\ 1 \quad (8)$$

3.1.5. Chebyshev map

The family of Chebyshev map is written as the following equation:

$$Y_{n+1} = \cos(kcos^{-1}(Y_n))\, Y \in (-1,1) \quad (9)$$

3.1.6. Sinusoidal map

This map can be represented by

$$Y_{n+1} = \mu Y_k^2 \sin(\pi Y_n) \quad (10)$$

3.1.7. Gauss map

The Gauss map is represented by:

$$Y_{n+1} = \begin{cases} 0 & Y_n = 0 \\ \frac{\mu}{Y_n} mod\ 1 & Y_n \neq 0 \end{cases} \quad (11)$$

3.1.8. Sinus map

Sinus map is formulated as follows:

$$Y_{n+1} = 2.3(Y_n)^{2\sin(\pi Y_n)} \quad (12)$$

3.1.9. Dyadic map

Also known as the dyadic map, bit shift map, 2*x* mod 1 map, Bernoulli map, doubling map or sawtooth map. Dyadic map can be formulated by a mod function:

$$Y_{n+1} = 2Y_n mod\ 1 \quad (13)$$

3.1.10. Singer map

Singer map can be written as:

$$Y_{n+1} = \mu(7.86Y_n - 23.31Y_n^2 + 28.75Y_n^3 - 13.3Y_n^4) \quad (14)$$

μ between 0.9 and 1.08

3.1.11. Tent map

This map can be defined by the following equation:

$$Y_{n+1} = \begin{cases} \mu Y_n Y_n < 0.5 \\ \mu(1-Y_n)Y_n \ge 0.5 \end{cases} \quad (15)$$

4. The Proposed Algorithm (CFPA) for Solving Fractional programming problems

In the proposed chaotic flower pollination algorithm, we used chaotic maps to tune the flower pollinationalgorithm parameter and improve the performance [19-20].
The steps of the proposedchaotic flower pollination algorithm for solving Fractional programming problemsare as follows:

```
Chaotic Flower pollination algorithm
Define Objective function $f(x)$, $x = (x_1, x_2, ..., x_d)$
Initialize a population of n flowers/pollen gametes with random solutions
Find the best solution Bin the initial population
Define a switch probability p,  where p has been chosen from chaotic sequence between [0, 1]
Tuning of parameters using chaotic maps (L, λ, U)
while (t <MaxGeneration)
for i = 1 : n (all n flowers in the population)
if rand <p,
Draw a (d-dimensional) step vector L which obeys a Lévy distribution
Global pollination via $x_i^{t+1} = x_i^t + L(B - x_i^t)$
else
Draw U from a uniform distribution in [0,1]
Do local pollination via $x_i^{t+1} = x_i^t + U(x_j^t - x_k^t)$
end if
Evaluate new solutions
If new solutions are better, update them in the population
end for
Find the current best solution B
end while
Output the best solution found
```

Figure 2. Pseudo code of the Chaotic Flower pollination algorithm

5. NumericalResults

Ten diverse problems were collected from literature[27], [16] to demonstrate the efficiency and robustness of solving FPPs. The obtained numerical results are compared to their relevance found in references; some examples were also solved using exact method $f_1$and f_3. Table 1shows they attained the comparison result. In these problems, the initial parameters are set at *n*= 50 and the number of iterations is set to *t* = 1000, the selected chaotic map for all problems is the logistic map, according to the following equation:

$$Y_{n+1} = \mu Y_n(1 - Y_n) \tag{16}$$

Clearly, $Y_n \in [0,1]$ under the conditions that the initial$Y_0 \in [0,1]$, where n is the iteration number and$\mu = 4$.The results of CFPA algorithm are conducted from 50 independent runs for each problem. The comparison between the results determined by the proposed approach and the compared algorithms are reported in Table 1. The results have demonstrated the superiority of the proposed approach to finding the optimal solution.
All the experiments were performed on a Windows 7 Ultimate 64-bit operating system; processor Intel Core i5 760 running at 2.81 GHz; 4 GB of RAMand code was implemented in MATLAB.

5.1. Test problem 1

This problem is defined as:

$$f_1: \max\ z = \frac{4x + 2y + 10}{x + 2y + 5}$$
$$\textit{subject to } x + 3y \le 30$$
$$-x + 2y \le 5;\quad x, y \ge 0$$

5.2. Test problem 2

Thetwo-dimensional Schaffer 2 function definedas:

$$0.5 + \frac{\sin\left(\sin\left(x_1^2 + x_2^2\right)\right) - 0.5}{\left(1 + 0.001\left(x_1^2 + x_2^2\right)\right)^2}$$
$$-100 \le x_i \le 100;\ i = 1, 2$$

5.3. Test problem 3

This problem is defined as:

$$f_3: \min\ z = \frac{x + y + 1}{2x - y + 3}$$
$$\textit{subject to } 0 \le x \le 1;\quad 0 \le y \le 1$$

5.4. Test problem 4

This problem is defined as:

$$f_4: \max\ z = \frac{8x + 7y - 2.33(9x^2 + 4y^2)^{0.5}}{20x + 12y - 2.33(3x^2 + 2xy + 4y^2)^{0.5}}$$
$$\textit{subject to } 2x + y \le 18$$
$$x + 2y \le 16;\quad x, y \ge 0$$

5.5. Test problem 5

The two-dimensional Schaffer 4 function defined as:

$$0.5 + \frac{\cos\left(\sin\left|x_1^2 + x_2^2\right|\right) - 0.5}{\left(1 + 0.001\left(x_1^2 + x_2^2\right)\right)^2}$$
$$-100 \le x_i \le 100;\ i = 1, 2$$

5.6. Test problem 6

This two-dimensional sineEnvelopefunction defined as:

$$f_6: \max z = -\sum_{i=1}^{n-1}\left[\frac{\sin^2\left(\sqrt{x_{i+1}^2 + x_i^2} - 0.5\right)}{\left(0.001\left(x_{i+1}^2 + x_i^2\right) + 1\right)^2} + 0.5\right]$$
$$-100 \le x_i \le 100;$$

5.7. Test problem 7

This problem is defined as:

$$f_7: \max z = \frac{-x^2y^{0.5} + 2xy^{-1} - y^2 + 2.8x^{-1}y + 7.5}{xy^{1.5} + 1} + \frac{y + 0.1}{x^2y^{-1} - 3x^{-1} + 2xy^2 + 9y^{-1} + 12}$$
$$\textit{subject to } 2x^{-1} + xy \le 4$$
$$x + 3x^{-1}y \le 5$$
$$x^2 - 3y^3 \le 2;\quad 1 \le x, y \le 3$$

5.8. Test problem 8

This problem is defined as:

$$f_8: \max\ z = \frac{37x + 73y + 13}{13x + 13y + 13} + \frac{63x - 18y + 39}{13x + 26y + 13}$$
$$\textit{subject to } 5x - 3y = 3$$
$$1.5 \le x \le 3$$
$$x, y \ge 0$$

5.9. Test problem 9

This problem is defined as:

$$f_9: \min\ z = \frac{2x + y}{x + 10} + \frac{2}{y + 10}$$
$$\textit{subject to } -x^2 - y^2 + 3 \le 0$$
$$-x^2 - y^2 + 8y - 14 \le 0$$
$$2x + y \le 6$$
$$3x + y \le 8$$
$$x - y \le 1$$
$$1 \le x \le 3;\quad 1 \le y \le 4$$

5.10. Test problem 10

This problem is defined as:

$$f_{10}:\max z=\left(\frac{13x+13y+13}{37x+73y+13}\right)^{-1.4}\times\left(\frac{64x-18y+39}{13x+26y+13}\right)^{1.2}-\left(\frac{x+2y+5v+50}{x+5y+5v+50}\right)^{0.5}\times\left(\frac{x+2y+4v+50}{5y+4v+50}\right)^{1.1}$$

$$\begin{aligned} \textit{subject to } & 2x+y+5v\le 10\\ & 5x-3y=3\\ & 1.5\le x\le 3; \qquad x,y,v\ge 0\end{aligned}$$

Table 1: Comparison results of the CFPA with other methods.

Test problem	Technique/Reference	Decision variable optimal value	Objective function value
f_1 (max)	C.C. Transformation “Exact Method”	(x*,y*)= (30,0)	**z*=3.714286**
	Dinkelbach algorithm “Exact Method”	(x*,y*)= (30,0)	**z*=3.714286**
	CFPA	(x*,y*)= (30,0)	z*= 3.7142857
f_2 (min)	C.C. Transformation “Exact Method”	none	none
	Dinkelbach algorithm “Exact Method”	none	none
	CFPA	(x*,y*)=(0.0056,0.0008)	**z*=6.25E-016**
f_3 (min)	C.C. Transformation “Exact Method”	(x*,y*)= (0,0)	z*=0.333
	Dinkelbach algorithm “Exact Method”	(x*,y*)= (0,0)	z*=0.333
	[28] “Neural Network”	(x*,y*)=(0.5,3)	z*=4.5
	CFPA	(x*,y*)= (0,0)	**z*=0.333**
f_4 (max)	C.C. Transformation “Exact Method”	none	none
	Dinkelbach algorithm “Exact Method”	none	none
	[29] “Goal Setting and Approximation”	(x*,y*)=(7.229,0)	z*=0.084
	CFPA	(x*,y*)= (1.0264,5.7391)	z*=0.3383
		(x*,y*)= (1.025,5.628)	**z*=0.3385**
f_5 (max)	C.C. Transformation “Exact Method”	none	none
	Dinkelbach algorithm “Exact Method”	none	none
	CFPA	(x*,y*)= (0,1.453)	**z*=0.290012**
f_6 (max)	C.C. Transformation “Exact Method”	none	none
	Dinkelbach algorithm “Exact Method”	none	none
	CFPA	(x*,y*)= (0,0)	**z*= 0**
f_7 (max)	C.C. Transformation “Exact Method”	none	none
	Dinkelbach algorithm “Exact Method”	none	none
	[30] “Global Optimization”	(x*,y*)= (1,1)	**z*=5.5167**
	CFPA	(x*,y*)= (1,1)	**z*=5.5167**
f_8 (max)	C.C. Transformation “Exact Method”	none	none
	Dinkelbach algorithm “Exact Metho	none	none
	[31] “Global Optimization”	(x*,y*)=(3,4)	**z*=5**
	CFPA	(x*,y*)= (3,4)	**z*=5**
f_9 (min)	C.C. Transformation “Exact Method”	none	none
	Dinkelbach algorithm “Exact Method”	none	none
	[6] “Global Ooptimization”	(x*,y*)=(1,1.4142)	**z*=0.48558**
	CFPA	(x*,y*)=(1,1.4)	z*=0.486
f_{10} (max)	C.C. Transformation “Exact Method”	none	none
	Dinkelbach algorithm “Exact Method”	none	none
	[1] “HS"	(x*,y*,v*)=(1.5,1.5,1.1)	z*=8.1207
	CFPA	(x*,y*,v*)=(1.5,1.5,0)	**z*=8.279**

The numerical results obtained using the proposed algorithm are compared to assorted exact methods and metaheuristic techniques as shown in table 1. Four exact methods were selected for solving the 10 benchmark functions and carrying out the comparison. The four methods are C.C. Transformation, Dinkelbach algorithm, Goal setting and approximation and global optimization. Neural network and harmony search are the other two metaheuristic intelligent techniques incorporated in the compare test. The some calculations are obtained out of the numerical solutions of all the ten functions. The obtained optimization value the proposed CFPA algorithm managed to explore new solution areas that benchmark problem results using exact methods couldn’t reach that could be clearly noticed from f_2 to f_{10} in table (1).The optimization value results for the rest functions indicates a better achievement for CFPA algorithm. Boldface figures in the table indicates the best result(s) among the algorithms. Figures 2,3 and 4. show that the proposed CFPA algorithm is able reach to global optimization in f_2,f_5, and f_6 though they have many local optimizations.

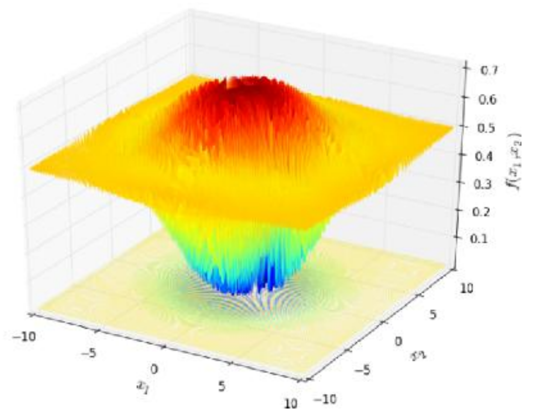

Figure 2: Two-dimensional Schaffer 2 function

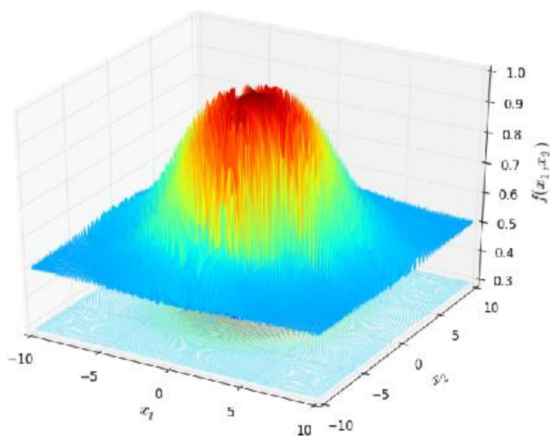

Figure 3: Two-dimensional Schaffer 4 function

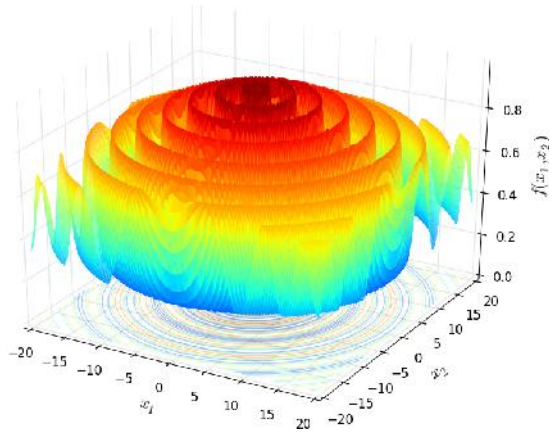

Figure 4: Two-dimensional Sine Envelope function

5.11. Corrugated Bulkhead Design

Corrugated bulkhead design [32]are often used in chemical tankers and product tankers in order to help facilities cargo tank washing effectively. This problem is as an example of minimum-weight design of the corrugated bulkheads for a tanker. Four design variables of the problem are width (*b*), depth (*h*), length (*l*), and plate thickness (*t*) for minimum-weight design of the corrugated bulkheads for a tanker, the mathematical formula for the optimization problem as follows:

$$Minimize: f\left(b,h,l,t\right)=\frac{5.885t\left(b+l\right)}{b+\sqrt{\left(l^2-h^2\right)}}$$

$$g_1=th\left(0.4b+\frac{1}{6}\right)-8.94\left(b+\sqrt{\left(l^2-h^2\right)}\right)\geq 0$$

$$g_2=th^2\left(0.2b+\frac{1}{12}\right)-2.2\left(8.94+\left(b+\sqrt{\left(l^2-h^2\right)}\right)\right)^{\frac{4}{3}}\geq 0$$

$$g_3=t-0.0156b-0.15\geq 0$$

$$g_4=t-0.0156l-0.15\geq 0$$

$$g_5=t-1.05\geq 0$$

$$g_6=l-h\geq 0,\quad 0\leq b,\ h,\ l\leq 100\ and\ 0\leq t\leq 5$$

Table 2: Comparison results of the corrugated bulkhead design example

	b(cm)	*h*(cm)	*l*(cm)	*t*(cm)	best
CFPA	57.69231	37.26590	57.69231	1.05	7.008391
CS	57.69231	37.36410	57.69231	1.05	7.01413
FA	57.69231	38.00090	57.69231	1.05	7.05219
PSO	57.69231	37.56410	57.69231	1.05	7.02594

The comparison result of minimum-weight and the statistical values of the best solution obtained by CFPA, CS,FA and PSO are given in Table (2). Accuracy performance is measured in terms of the best, mean, and standard deviation values of the solutions obtained by 20 independent runs. The best minimum-weight in this study is 7.008391 with thickness 1.05 cm.

5.12. Design of a Gear Train

The below figure (5) shows the gear train problem[16]. A gear ratio between the driver and driven shafts must be achieved when designing a compound gear train. The gear ratio for gear train is defined as the ratio of the angular velocity of the output shaft to that of the input shaft. It is desirable to produce a gear ratio as close as possible to 1/6.931. For each gear, the number of teeth must be between 12 and 60. The design variables T_a, T_b, T_d, and T_f are the numbers of teeth of the gears *a*, *b*, *d* and *f*, respectively, which must be integers.

$$\vec{x}=\left(T_d,T_b,T_a,T_f\right)^T=\left(x_1,x_2,x_3,x_4\right)^T$$

The optimization problem is expressed as:

$$\min\ z=\left(\frac{1}{6.931}-\frac{\lfloor T_d\rfloor\lfloor T_b\rfloor}{\lfloor T_a\rfloor\lfloor T_f\rfloor}\right)^2=\left(\frac{1}{6.931}-\frac{\lfloor x_1\rfloor\lfloor x_2\rfloor}{\lfloor x_3\rfloor\lfloor x_4\rfloor}\right)^2$$

$$subject\ to\quad 12\leq x_i\leq 60\qquad i=1,2,3,4$$

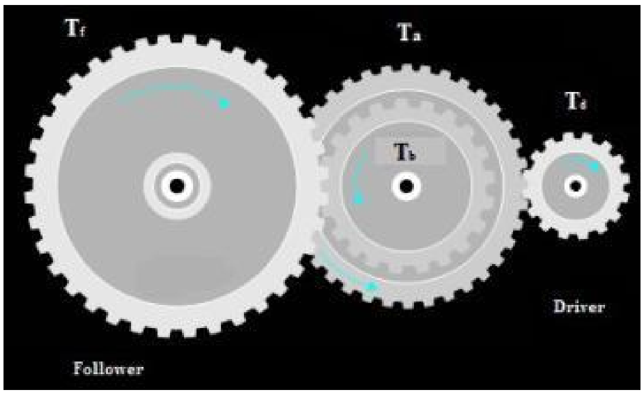

Figure 5: A gear train.

The constraint ensures that the error between obtained gear ratio and the desired gear ratio is not more than the 50% of the desired gear ratio.

Table 3: Comparison results of the CFPA, CS, FA and PSO

	CFPA	CS	FA	PSO
Best	**2.7E-012**	2.70009E-012	2.7E-012	2.700857E-012
Mean	2.7E-012	1.04E-010	5.1314E-012	1.1371E-011
StdDev.	0.00E+00	2.7E-010	7.6868E-012	1.01307E-011

The comparison resulted obtained by the CFPA, CA, FA and PSO algorithms are given in Table (3), The comparison in terms of the best, error, mean, standard deviation values, these values where obtained out of 20 independent runs. The result indicates a better achievement for CFPA, with an objective function value of 2.7 E-012.

6. Conclusion

The paper presents a new approach to solve FPPs based on Flower pollination algorithm with chaos. Ten-benchmark problem weresolvedusingthe proposed algorithm and many other previous approaches.The results employing the proposed algorithmwerecompared with the other exact and metaheuristic approachpreviously used for handling FPPs. The algorithm proved their effectiveness, reliability and competences insolving different FPPs. The proposed algorithm managed tosuccessfullysolve large-scale FPPs with an optimal solution at a finite point and an unbounded constraint set. The features and capabilitiesof the proposed algorithm was more evident when dealing with large-scale problems and a solution is a regular space.Thecomputational results proved that CFPA turned out to besuperior to other algorithms for all the accomplished testsyielding a higher and much faster growing mean fitness atless computational time. The proposed algorithm can extend to handle multiobjective fractional optimization.

References

[1] M. Jaberipour and E. Khorram, "Solving the sum-of-ratios problems by a harmony search algorithm," *Journal of computational and applied mathematics*, vol. 234, pp. 733–742, 2010.

[2] H. Wolf, "A parametric method for solving the linear fractional programming problem," *Operations Research*, vol. 33, pp. 835–841, 1985.

[3] A. Charnes and W. Cooper, "An explicit general solution in linear fractional programming," *Naval Research Logistics Quarterly*, vol. 20, pp. 449–467, 1973.

[4] M. Hosseinalifam, "A Fractional Programming Approach for Choice-Based Network Revenue Management," UNIVERSITE DE MONTREAL, 2009.

[5] I. Stancu-Minasian, *Fractional programming: theory, methods and applications*, vol. 409. Kluwer academic publishers Dordrecht, 1997.

[6] H. Jiao, Z. Wang, and Y. Chen, "Global optimization algorithm for sum of generalized polynomial ratios problem," *Applied Mathematical Modelling*, vol. 37, pp. 187–197, 2013.

[7] P. Shen, Y. Chen, and Y. Ma, "Solving sum of quadratic ratios fractional programs via monotonic function," *Applied Mathematics and Computation*, vol. 212, pp. 234–244, 2009.

[8] A. Sameeullah, S. D. Devi, and B. Palaniappan, "Genetic algorithm based method to solve linear fractional programming problem," *Asian Journal of Information Technology*, vol. 7, pp. 83–86, 2008.

[9] H. I. Calvete, C. Galé, and P. M. Mateo, "A genetic algorithm for solving linear fractional bilevel problems," *Annals of Operations Research*, vol. 166, pp. 39–56, 2009.

[10] S. Bisoi, G. Devi, and A. Rath, "Neural Networks for Nonlinear Fractional Programming," *International Journal of Scientific & Engineering Research, Volume 2, Issue 12, December-2011*, vol. 2, 12, pp. 1–5, 2011.

[11] L. Xiao, "Neural Network Method for Solving Linear Fractional Programming," in *Computational Intelligence and Security (CIS), 2010 International Conference on*, 2010, pp. 37–41.

[12] A. Pal, S. Singh, and K. Deep, "Solution of fractional programming problems using PSO algorithm," in *Advance Computing Conference (IACC), 2013 IEEE 3rd International*, 2013, pp. 1060–1064.

[13] I. M. Hezam and O. A. Raouf, "Employing Three Swarm Intelligent Algorithms for Solving Integer Fractional Programming Problems," *International Journal of Scientific and Engineering Research (IJSER)*, vol. 4, pp. 191–198, 2013.

[14] M. Abdel-Baset and I.M. Hezam, "An Effective Hybrid Flower Pollination and Genetic Algorithm for Constrained Optimization Problems," *Advanced Engineering Technology and Application An International Journal*, vol. 4, 2015, pp. 27 – 27–34.

[15] M. Abdel-Baset and I. Hezam "An Improved Flower Pollination Algorithm for Ratios optimization Problems," Applied Mathematics & Information Sciences Letters An International Journal, vol. 3, No. 2, pp. 83–91,2015

[16]O. Raouf, I. El-henawy, and M. Abdel-Baset. "A novel hybrid flower pollination algorithm with chaotic harmony search for solving Sudoku puzzles." International Journal of Modern Education and Computer Science vol. 3, pp. 38-44, 2014.

[17] O. Raouf, M. A. Baset, I. Elhenawy, "A New Hybrid Flower Pollination Algorithm for Solving Constrained Global Optimization Problems", International Journal of Applied Operational Research Vol. 4, No. 2, pp. 1-13, 2014.

[18]O.Raouf, M.A. Baset, I. Elhenawy, "chaotic Harmony Search Algorithm with Different Chaotic Maps for Solving Assignment Problems", International Journal of Computational Engineering & Management,Vol. 17, pp. 10-15, 2014.

[19] L. M. Pecora and T. L. Carroll, "Synchronization in chaotic systems," *Physical review letters*, vol. 64, p. 821, 1990.

[20] O. Abdel-Raouf, M. Abdel-Baset, and I. El-henawy, "Improved Harmony Search with Chaos for Solving Linear Assignment Problems", International Journal of Intelligent Systems and Applications, vol. 6,pp. 55-61 ,2014.

[21] A. Gandomi, X.-S. Yang, S. Talatahari, and A. Alavi, "Firefly algorithm with chaos," *Communications in Nonlinear Science and Numerical Simulation*, vol. 18, pp. 89–98, 2013.

[22] J. Mingjun and T. Huanwen, "Application of chaos in simulated annealing," *Chaos, Solitons & Fractals*, vol. 21, pp. 933–941, 2004.

[23] O. Abdel-Raouf, M. Abdel-Baset, and I. El-Henawy, "An Improved Chaotic Bat Algorithm for Solving Integer Programming Problems," *International Journal of Modern Education and Computer Science (IJMECS)*, vol. 6, p. 18, 2014.

[24] O. Abdel-Raouf, M. Abdel-Baset, and I. El-henawy, "An Improved Flower Pollination Algorithm with Chaos," *I.J. Education and Management Engineering*, vol. 2, pp. 1–8, 2014.

[25] O. Abdel-Raouf, M. Abdel-Baset, and I. El-henawy, "Chaotic firefly algorithm for solving definite integral", International Journal of Information Technology and Computer Science, vol. 6,pp. 19-24,2014.

[26] O. A. Raouf, M. A. Baset, I. M. Elhenawy, "Improved harmony search algorithm with chaos for solving definite integral", International Journal of Operational Research,Vol. 21, No. 2, pp. 252-261, 2014.

[27] I. M. Hezam, O. A. Raouf, and M. M. Hadhoud, "A New Compound Swarm Intelligence Algorithms for Solving Global Optimization Problems," *INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY*, vol. 10, pp. 2010–2020, 2013.

[28] Q.-J. Zhang and X. Q. Lu, "A Recurrent Neural Network for Nonlinear Fractional Programming," *Mathematical Problems in Engineering*, vol. 2012, 2012.

[29] Y. Z. Mehrjerdi, "Solving fractional programming problem through fuzzy goal setting and approximation," *Applied Soft Computing*, vol. 11, pp. 1735–1742, 2011.

[30] P. Shen, Y. Ma, and Y. Chen, “Global optimization for the generalized polynomial sum of ratios problem,” *Journal of Global Optimization*, vol. 50, pp. 439–455, 2011.

[31] C.-F. Wang and P.-P. Shen, “A global optimization algorithm for linear fractional programming,” *Applied Mathematics and Computation*, vol. 204, pp. 281–287, 2008.

[32] I. M. H. MOHAMED ABDEL-BASET, “AN IMPROVED FLOWER POLLINATION ALGORITHM BASED ON SIMULATED ANNEALING FOR SOLVING ENGINEERING OPTIMIZATION PROBLEMS,” *Asian Journal of Mathematics and Computer Research*, vol. 3, pp. 149–170, 2015.

[33] I. M. Hezam, M. Abd-ElBaset and I. Selem, “Cuckoo Search Algorithm for Stellar Population Analysis of Galaxies”, International Journal of Information Technology and Computer Science, vol. 7, pp.29-33,2015.