Applied Mathematics & Information Sciences Letters
An International Journal

http://dx.doi.org/10.12785/amisl/030206

An Improved Flower Pollination Algorithm for Ratios Optimization Problems

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

[1] Department of Operations Research, faculty of Computers and Informatics, Zagazig University, El-Zera Square, Zagazig, Sharqiyah, Egypt.
[2] Department of computer, Faculty of Education, Ibb University, Ibb city, Yemen.

Received: 21 Feb. 2015, Revised: 22 Mar. 2015, Accepted: 24 Mar. 2015.
Published online: 1 May 2015.

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 based on the Flower Pollination Algorithm combined with Chaos Theory (IFPCH) to solve Ratios Optimization Problems (ROPs). The proposed algorithm is tested using several ROP benchmark. The test aims to prove the capability of the IFPCH to solve any type of ROPs. The solution results employing the IFPCH algorithm are compared with a number of exact and metaheuristic solution methods used for handling ROPs. Numerical examples are given to show the feasibility, effectiveness, and robustness of the proposed algorithm. The results obtained using IFPCH revealed the superiority of the proposed technique among others in computational time.

Keywords: Flower Pollination Algorithm; Nature-Inspired Algorithm; Optimization; Chaos; Ratios Optimization Problems.

1 Introduction

The general ratios optimization problems ROP mathematical model [1] is:

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

Where $f_i(x)$, $g_i(x)$, are supposed to be continuous functions, S is compact.

Ratios optimization problems of the form (1) arises 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 or at least concave-convex fractional programming.

Ratios optimization problems(fractional programming) 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 methods to solve ratios optimization problems can be broadly classified into exact (traditional) and heuristics approaches. Some examples of traditional approaches is that of ([2] who introduced the parametric approach, [3]solved the linear fractional programming problems by converting FPP into an equivalent linear programming problem and solved it using already existing standard algorithms for LPP, [4–7] 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 LFP problem to the solution of a sequence of LP problems. Isbell Marlow method, Martos' Algorithm, Cambini–Martein's Algorithm, Bitran and Novae's Method, Swarup's Method, Harvey M. Wagner and John S. C. Yuan,

[*]Corresponding author e-mail: analyst_mohamed@yahoo.com

Hasan, B.M., and Acharjee, S., developed a new method for solving FLPP based on the idea of solving a sequence of auxiliary problems so that the solutions of the auxiliary problems converge to the solution of the FPP.Moreover, there are many recent approaches employing traditional mathematical methods for solving the ratios optimization FPP such as: [8–10].

A few studies in recent years used heuristics approaches to solve ratios optimization problems. [11] 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 pair wise solution points are used for crossover and a new set of solution points is obtained. These steps are repeated for a certain number of generations and the best solution for the given problem is obtained. [12] 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. [13] 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 multiobjective minimax 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. [14] 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. [15] Used particle swarm optimization algorithm for solving fractional programming problems.[16–19] 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. Its evolutionary characteristics can be used to design new optimization algorithms. The algorithm obtained good results were dealing with lower-dimensional optimization problems, 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 chaos to 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 is briefly introduced. In section 3, the proposed algorithm is described, while the results are discussed in section 4. Finally, conclusions are presented in section 5.

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[20].Therefore, Rule 1 and flower constancy can be represented mathematically as:

$$x_i^{t+1} = x_i^t + \gamma L(\lambda)(x_i^t - B) \qquad (2)$$

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:

$$L \sim \frac{\lambda\Gamma(\lambda)\sin(\pi\lambda/2)}{\pi}\frac{1}{S^{1+\lambda}}, (S >> S_0 > 0) \qquad (3)$$

Here, $\Gamma(\lambda)$ 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:

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

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 = 0.5as an initially value. A preliminary parametric showed that p = 0.8 might work better for most applications[20].

```
Flower pollination algorithm
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 $p \in [0, 1]$
Define a stopping criterion (either a fixed number of generations/iterations or accuracy)
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 L´evy 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
```

Fig. 1 Pseudo code of the Flower pollination algorithm.

3 The Proposed Algorithm (IFPCH) for Solving Ratios Optimization Problems

Generating random sequences with a longer period and good consistency is very important for easily simulating complex phenomena, sampling, numerical analysis, decision making and especially in heuristic optimization[21]. Its quality determines the reduction of storage and computation time to achieve a desired accuracy[22]. 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[23–28]. Applications of chaos in several disciplines including operations research, physics, engineering, economics, biology, philosophy and computer science[29–32].

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 [33], [34], [30], [35], [36]. Using chaotic sequences in flower pollination algorithm can be helpful to improve the reliability of the global optimality, and they 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) [22]. 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 [37–40], [36]. To achieve this issue, herein one-dimensional, non-invertible 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:

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

3.1.2 The Sine map

The Sine map is written as the following equation:

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

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:

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

3.1.5 Chebyshev map

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

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

3.1.6 Sinusoidal map

This map can be represented by:

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

3.1.7 Gauss map

The Gauss map is represented by:

$$Y_{n+1} = \begin{cases} 0 & Y_n = 0 \\ \dfrac{\mu}{Y_n} mod\ 1 & Y_n \neq 0 \end{cases} \tag{11}$$

3.1.8 Sinus map

Sinus map is formulated as follows:

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

3.1.9 Dyadic map

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

$$Y_{n+1} = 2Y_n mod\ 1 \tag{13}$$

3.1.10 Singer map

Singer map can be written as:

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

μ between 0.9 and 1.08

3.1.11 Tent map

This map can be defined by the following equation:

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

3.2 Handling Constraints

One of the well-known techniques of handling constraints is using penalty function, which transforms constrained problem into unconstrained ones, consisting of a sum of the objective and the constraints weighted by penalties. By using penalty function methods, the objectives are inclined to guide the search toward the feasible solutions. Hence, in this paper the corresponding objective function used in is defined and described as:

$$\min\ F(x) = f(x) + \lambda \sum_{n=1}^{K} \max(0, g_n)$$

Where $f(x)$ is the objective function for assignment problem, λ is the penalty coefficient and it is set to a value of 10^{11} in this paper, K is the number of constraints and g_n the constraints of the problem.

In the proposed chaotic flower pollination algorithm, we used chaotic maps to tune the flower pollination algorithm parameter and improve the performance [21], [22].

The steps of the proposed chaotic flower pollination algorithm for solving ratios optimization problems are as follows:

Step 1 define the objective function and initializes a population then find the best solution *B* in the initial population.

Step 2 Calculate *p* by the selected chaotic maps.

Step 3 *If* (*rand* <*p*) then global pollination via $x_i^{t+1} = x_i^t + (f\gamma)L(\lambda)(x_i^t - B)$

else do local pollination via selected chaotic map.

Step 4 Evaluate new solutions if better, update them in the population.

Step 5 Find the current best solution *B*.

Step 6 Output the best solution found.

4 Numerical Results

Ten diverse problems were collected from literature[41], [19] to demonstrate the efficiency and robustness of solving FFPs. 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 IFPCH 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 RAM and code was implemented in MATLAB.

4.1. Test problem 1

This problem is defined as:

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

4.2. Test problem 2

The two-dimensional Schaffer 2 function defined as:

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

4.3. Test problem 3

This problem is defined as:

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

4.4. Test problem 4

This problem is defined as:

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

4.5. Test problem 5

The two-dimensional Schaffer 4 function defined as:

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

4.6. Test problem 6

This two-dimensional sine Envelope function defined as:

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

4.7. Test problem 7

This problem is defined as:

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

4.8. Test problem 8

This problem is defined as:

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

4.9. Test problem 9

This problem is defined as:

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

4.10. Test problem 10

This problem is defined as:

$$f_{10}:\max z=\left(\frac{13x+13y+13}{37x+73y+13}\right)^{-1.4}\times\left(\frac{64x-18y+39}{13x+26y+13}\right)^{1.2}-\left(\frac{x+2y+5v+50}{x+5y+5v+50}\right)^{0.5}\times\left(\frac{x+2y+4v+50}{5y+4v+50}\right)^{1.1}$$
$$\textit{subject to } 2x+y+5v\le 10$$
$$5x-3y=3$$
$$1.5\le x\le 3;\quad x,y,v\ge 0$$

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

Test problem	**Technique/Reference**	**Decision variable optimal value**	**Objective function value**
f_1 **(max)**	C.C. Transformation “Exact Method” Dinkelbach algorithm “Exact Method” IFPCH	(x*,y*)= (30,0) (x*,y*)= (30,0) (x*,y*)= (30,0)	z*=3.714286 z*=3.714286 z*= 3.7142857
f_2 **(min)**	C.C. Transformation “Exact Method” Dinkelbach algorithm “Exact Method” IFPCH	none none (x*,y*)=(0.0056,0.0008)	none none z*=6.25E-016
f_3 **(min)**	C.C. Transformation “Exact Method” Dinkelbach algorithm “Exact Method” [42] “Neural Network” IFPCH	(x*,y*)= (0,0) (x*,y*)= (0,0) (x*,y*)=(0.5,3) (x*,y*)= (0,0)	z*=0.333 z*=0.333 z*=4.5 z*=0.333
f_4 **(max)**	C.C. Transformation “Exact Method” Dinkelbach algorithm “Exact Method” [43] “Goal Setting and Approximation” IFPCH	none none (x*,y*)=(7.229,0) (x*,y*)= (1.0264,5.7391) (x*,y*)= (1.025,5.628)	none none z*=0.084 z*=0.3383 z*=0.3385
f_5 **(max)**	C.C. Transformation “Exact Method” Dinkelbach algorithm “Exact Method” IFPCH	none none (x*,y*)= (0,1.453)	none none z*=0.290012
f_6 **(max)**	C.C. Transformation “Exact Method” Dinkelbach algorithm “Exact Method” IFPCH	none none (x*,y*)= (0,0)	none none z*= 0
f_7 **(max)**	C.C. Transformation “Exact Method” Dinkelbach algorithm “Exact Method” [44] “Global Optimization” IFPCH	none none (x*,y*)= (1,1) (x*,y*)= (1,1)	none none z*=5.5167 z*=5.5167
f_8 **(max)**	C.C. Transformation “Exact Method” Dinkelbach algorithm “Exact Metho [45] “Global Optimization” IFPCH	none none (x*,y*)=(3,4) (x*,y*)= (3,4)	none none z*=5 z*=5
f_9 **(min)**	C.C. Transformation “Exact Method” Dinkelbach algorithm “Exact Method” [9] “Global Ooptimization” IFPCH	none none (x*,y*)=(1,1.4142) (x*,y*)=(1,1.4)	none none z*=0.48558 z*=0.486
f_{10} **(max)**	C.C. Transformation “Exact Method” Dinkelbach algorithm “Exact Method” [1] “HS" IFPCH	none none (x*,y*,v*)=(1.5,1.5,1.1) (x*,y*,v*)=(1.5,1.5,0)	none none z*=8.1207 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 IFPCH 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 IFPCH algorithm. Boldface figures in the table indicates the best result(s) among the algorithms. Figures 2, 3 and 4. show that the proposed IFPCH algorithm is able reach to global optimization in f_2,f_5, and f_6 though they have many local optimizations.

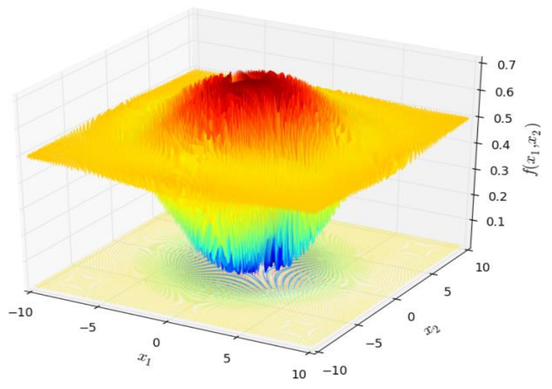

Fig. 2: Two-dimensional Schaffer 2 function.

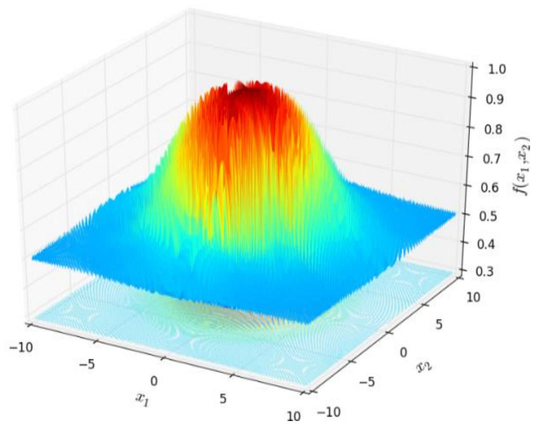

Fig. 3: Two-dimensional Schaffer 4 function.

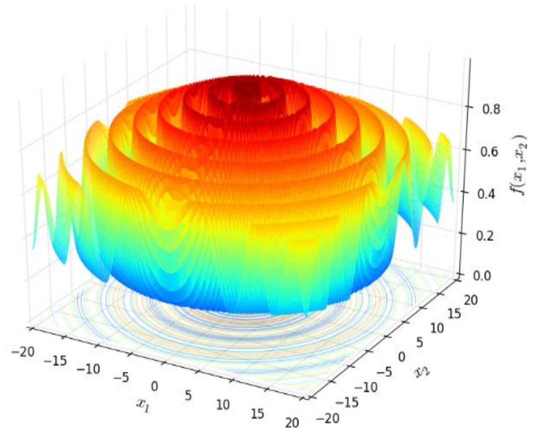

Fig. 4: Two-dimensional Sine Envelope function.

5 Conclusions

The paper presents a new approach to solve ROPs based on Flower pollination algorithm with chaos. Ten-benchmark problem were solved using the proposed algorithm and many other previous approaches. The results employing the proposed algorithm were compared with the other exact and metaheuristic approach espreviously used for handling ROPs. The algorithm proved their effectiveness, reliability and competences in solving different ROP. The proposed algorithm managed to successfully solve large-scale ROP with an optimal solution at a finite point and an unbounded constraint set. The features and capabilities of the proposed algorithm was more evident when dealing with large-scale problems and a solution is a regular space. The computational results proved that IFPCH turned out to be superior to other algorithms for all the accomplished tests yielding a higher and much faster growing mean fitness at less computational time.

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] T. B. Farag, "A Parametric Analysis on Multicriteria Integer Fractional Decision-Making Problems," Faculty of Science, Helwan University, (2012).

[5] M. B. Hasan and S. Acharjee, "Solving LFP by Converting it into a Single LP," *International Journal of Operations Research*, vol. **8**, pp. 1–14, (2011).

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

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

[8] M. Dür, C. Khompatraporn, and Z. B. Zabinsky, "Solving fractional problems with dynamic multistart improving hit-and-run," *Annals of Operations Research*, vol. **156**, pp. 25–44, (2007).

[9] 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).

[10] 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).

[11] 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).

[12] 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).

[13] 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).

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

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

[16] 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).

[17] I. M. Hezam and O. A. Raouf, "Particle Swarm Optimization Approach For Solving Complex Variable Fractional Programming Problems," *International Journal of Engineering*, vol. **2**, (2013).

[18] I. M. Hezam and M. M. H. Osama Abdel Raouf, "Solving Fractional Programming Problems Using Metaheuristic Algorithms Under Uncertainty," *International Journal of Advanced Computing*, vol. **46**, pp. 1261–1270, (2013).

[19] A.-R. Osama, A.-B. Mohamed and E.-h. Ibrahim, "A Novel Hybrid Flower Pollination Algorithm with Chaotic Harmony Search for Solving Sudoku Puzzles," International Journal of Engineering Trends and Technology (IJETT), vol. **7**, no. 3, pp. 126-132, January (2014).

[20] X.-S. Yang, "Flower pollination algorithm for global optimization," in *Unconventional Computation and Natural Computation*, Springer, pp. 240–249, (2012).

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

[22] D. Yang, G. Li, and G. Cheng, "On the efficiency of chaos optimization algorithms for global optimization," *Chaos, Solitons & Fractals*, vol. **34**, pp. 1366–1375, (2007).

[23] B. Alatas, "Chaotic bee colony algorithms for global numerical optimization," *Expert Systems with Applications*, vol. **37**, pp. 5682–5687, (2010).

[24] 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).

[25] A. H. Gandomi, G. J. Yun, X.-S. Yang, and S. Talatahari, "Chaos-enhanced accelerated particle swarm optimization," *Communications in Nonlinear Science and Numerical Simulation*, vol. **18**, pp. 327–340, (2013).

[26] W. Gong and S. Wang, "Chaos ant colony optimization and application," in *Internet Computing for Science and Engineering (ICICSE), 2009 Fourth International Conference on*, pp. 301–303, (2009).

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

[28] L. dos Santos Coelho and V. C. Mariani, "Use of chaotic sequences in a biologically inspired algorithm for engineering design optimization," *Expert Systems with Applications*, vol. **34**, pp. 1905–1913, (2008).

[29] 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).

[30] D. He, C. He, L.-G. Jiang, H.-W. Zhu, and G.-R. Hu, "Chaotic characteristics of a one-dimensional iterative map with infinite collapses," *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, vol. **48**, pp. 900–906, (2001).

[31] R. C. Hilborn, *Chaos and nonlinear dynamics*, vol. 2. Oxford University Press New York, (1994).

[32] M. S. Tavazoei and M. Haeri, "Comparison of different one-dimensional maps as chaotic search pattern in chaos optimization algorithms," *Applied Mathematics and Computation*, vol. **187**, pp. 1076–1085, (2007).

[33] 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).

[34] A. Erramilli, R. Singh, and P. Pruthi, *Modeling packet traffic with chaotic maps*. Citeseer, (1994).

[35] R. M. May and others, "Simple mathematical models with very complicated dynamics," *Nature*, vol. **261**, pp. 459–467, (1976).

[36] A. Wolf, "Quantifying chaos with Lyapunov exponents," *Chaos*, pp. 273–290, (1986).

[37] R. Barton, "Chaos and fractals," *The Mathematics Teacher*, pp. 524–529, (1990).

[38] R. L. Devaney, "An introduction to chaotic dynamical systems," (2003).

[39] C. Letellier, *Chaos in nature*, vol. 81. World Scientific, (2013).

[40] E. Ott, *Chaos in dynamical systems*. Cambridge university press, (2002).

[41] O. A. Raouf, M. A. Baset, I. M. 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, Spring (2014).

[42] Q.-J. Zhang and X. Q. Lu, “A Recurrent Neural Network for Nonlinear Fractional Programming,” *Mathematical Problems in Engineering*, vol. **2012**, (2012).

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

[44] 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).

[45] 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).