Received ; | Revised ; | Accepted
DOI: xxx/xxxx

RESEARCH ARTICLE

Research on nature-inspired metaheuristics for optimal control of fractional differential equation

Asyieh Ebrahimzadeh[1] | Raheleh Khanduzi*[2] | Samaneh Panjeh Ali Beik[3]

[1]Farhangian university, Department of Mathematics Education, Tehran, Iran
[2]Gonbad Kavous University, Department of Mathematics and Statistics, Golestan, Iran
[3]Islamic Azad University, Young Researchers and Elite Club, Karaj Branch, Iran

Correspondence
*Raheleh Khanduzi. Email: khanduzi@gonbad.ac.ir

Present Address
Department of Mathematics and Statistics, Gonbad Kavous University, P.O. Box 49717-99151, Gonbad Kavous, Golestan, Iran

Summary

This study offers a general formulation for a class of fractional optimal control problems (FOCPs) where the performance index is expressed as a function of both control and state variables and the dynamic control system depends on Caputo fractional derivatives. The operational matrices of fractional Riemann-Liouville integration for Bernoulli polynomials and properties of Bernoulli polynomials are utilized to reduce the given optimization problems to the nonlinear programming problem (NLP) by solving of which we can approximate the optimal solution of FOCP. By implementing three metaheuristic approaches called multi-verse optimizer (MVO), moth-flame optimization (MFO), and whale optimization algorithm (WOA), the NLP is solved and the best approximation solution of FOCP is obtained. A survey on the superiority and the efficiency between these methods are considered by applying three numerical examples. Comprehensive analysis reveals that the MFO considerably solves these examples. Moreover, the profits and advantages of preference with its precision are demonstrated numerically. Simulation results obviously show that the objective functional value obtained by MFO effectively decreased on three illustrative examples in comparison with MVO and WOA.

KEYWORDS:
Optimal control , Fractional differential equation , Bernoulli polynomials , Moth-flame optimization , Multi-verse optimizer , Whale optimization algorithm

1 | INTRODUCTION

Exact modeling of various dynamical systems leads to a set of fractional differential equations. Interested reader in fractional calculus and fractional differential equations can study [2,12] and references therein. The main purpose of this paper is to introduce some efficient approaches for solving a class of OCPs with the Caputo fractional derivative in a dynamical system [12]:

$$\mathcal{J} = \int_0^1 \mathcal{F}(t, x(t), u(t))dt, \tag{1}$$

subject to

$$\mathcal{D}^{\alpha} x(t) = \mathcal{G}(t, x(t)) + \beta(t)u(t), \tag{2}$$

with initial conditions

$$x(0) = x_0, \quad x'(t_0) = x_1, \cdots, x^{[q]-1}(t_0) = x_{[q]-1}, \tag{3}$$

where f, g and β are smooth function of their arguments. We suppose without loss of generality that the interval of integration in Equation (1) is $[0, 1]$, since any finite interval $[t_0, t_f]$ can be transformed to interval $[0, 1]$ by a linear transformation. Analytical discussions about the existence and uniqueness of the optimal control problem (1)-(3) can be found in [15].

Because most of FOCPs do not chiefly have exact analytic solutions, this highlights the importance of accurate approximated and numerical methods. The study of fractional variational problems with derivatives in the sense of Caputo is a recent subject. All methods presented for solving OCPs are partitioned into two classes as direct and indirect in which the former methods describe the continuous FOCPs to a finite-dimensional NLP and others are based on the necessary optimality conditions of a OCP. An up-to-date bibliography on various numerical approaches for solving FOCPs was lately reported by [4] and [7].

The direct computational method utilized in this essay consists of reducing FOCP to a NLP. Firstly, the fractional state rate $\mathcal{D}^{\alpha}x(t)$ and control function $u(t)$ by an orthogonal polynomial with unknown coefficients. Approximations attained by this technique satisfy all the initial conditions of the problem which is a substantial property. Then the operational matrix of fractional integration have been utilized to the achieve a NLP, by solving of which we can approximate the optimal solution of the main FOCP up to (3) in terms of the unknown coefficients.

At the final phase, we utilize three metaheuristic algorithms, called multi-verse optimizer (MVO), moth-flame optimization (MFO), and whale optimization algorithm (WOA) to gain unknown coefficients in the resulted NLP. An application of the new method based on Bernoulli polynomials and metaheuristic algorithms on three numerical examples is presented. In what follows, some descriptions on the advantages of the MVO, MFO and WOA are mentioned. It is valuable implying that, the metaheuristic algorithms are executed as the efficient approaches for solving large-scale nonlinear programming problems; see Mirjalili et al. [21], Mirjalili [22], Mirjalili and Lewis [23] and several studies in the literature regarding the application of the MVO, MFO and WOA; see, e.g., Faris et al. [24]; Fathy and Rezk [25]; Jangir et al. [26]; YÄśldÄśz and YÄśldÄśz [27]; Mei et al. [28]; Khalilpourazari and Khalilpourazary [29]; Aljarah et al. [30]; Mafarja and Mirjalili [31]; Abdel-Basset et al. [32]. The FOCP is generally converted to a large-scale NLP; therefore, the classical optimization methods might have defects for solving this problem due to the local optimum, considerable time and challenging implementation, while the metaheuristic approaches are run efficiently and can gain reasonable and satisfactory solutions concerning execution time and precision. These verifiable details as well as the following reasons stimulate us to utilize the MVO, MFO and WOA to solve NLPs governed from FOCP:

- *Global search:* The predominant advantage of the MVO, MFO and WOA is the lower probability of fall in local solutions compared to the exact-solution-based methods. So, the MVO, MFO and WOA are warranted to converge to the global solution.

- *Simplicity:*These metaheuristic algorithms in this study followed a simple structure and had been inspired from simple ideas in cosmology (MVO), navigation method of moths (MFO), and social behavior of humpback whales(WOA).

- *Black boxes:* The MVO, MFO and WOA algorithms dealt with problems as black boxes, therefore they did not require derivative information of the search region against classical and local search techniques.

- *Flexible:* The MVO, MFO and WOA approaches were exceedingly flexible, implying that they were easily applicable to solve different NLPs without fundamental modifications. Actually, the problem statement become more essential and meaningful than the optimizer when using the MVO, MFO and WOA algorithms.

- *Effectiveness:* Using the value of objective functional, we would be able to deduce the validity of three metaheuristic algorithms. The results of Tables 1 , 2 and 3 indicated that the MFO was the most effective approach in comparison with MVO and WOA with respect to exact solutions and maximum absolute errors.

- *Large-scale setting:* Our NLP was a large-sized problem. The classical optimization algorithms were satisfactory for small- or medium-sized problems. As reported in Tables 1 , 2 and 3 , the MVO, MFO and WOA algorithms were appropriate for large-sized problems.

- *Application:* To the best of our knowledge, MVO, MFO and WOA have not been executed to solve FOCP in the literature.

This paper is unified as follows. In section 2, firstly, some definitions and mathematical preliminaries of fractional calculus needed for our consequent development are described and then the Bernoulli operational matrices of the fractional integration is given. The detailed implementation of direct methods and three metaheuristic to solve NLPs governed from FOCP is presented in section 3. In Section 4, we report our numerical findings and demonstrate the accuracy of the propounded numerical scheme by considering some numerical examples. Furthermore, comprehensive comparisons between the MFO, MVO and WOA are represented to show the applicability of the developed approach. Lastly, in section 5, the paper brings an end with a concise conclusion and some remarks.

2 | PRELIMINARIES

This section consists of some principle definitions and properties of Bernoulli orthogonal polynomials, fractional calculus and function approximation.

2.1 | Bernoulli polynomials and their properties

The Bernoulli polynomials of order i are specified by a series[16]

$$\mathcal{B}_i(t) = \sum_{k=0}^{i} \binom{i}{k} \alpha_k t^{i-k} = \sum_{k=0}^{i} \binom{i}{k} \alpha_{i-k} t^i \tag{4}$$

where $\alpha_i,\ i = 0, 1, \cdots, m$ are Bernoulli numbers. These numbers are a sequence of signed rational numbers which arise in the series expansion of trigonometric functions[20] and can be defined by the identity

$$\frac{t}{e^t - 1} = \sum_{i=0}^{\infty} \alpha_i \frac{t^i}{i!}.$$

The first few Bernoulli numbers are

$$\alpha_0 = 1, \quad \alpha_1 = -\frac{-1}{2}, \quad \alpha_2 = \frac{1}{6}, \quad \alpha_4 = \frac{-1}{30} \tag{5}$$

with $\alpha_{2i+1} = 0, \quad i = 1, 2, 3, \cdots$. In[16], Bernoulli numbers have been obtained by the following determinantal definition

$$\alpha_0 = 0$$

$$\alpha_i = \frac{(-1)^i}{(i-1)!} \begin{vmatrix} \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \cdots & \frac{1}{i} & \frac{1}{i+1} \\ 1 & 1 & 1 & \cdots & 1 & 1 \\ 0 & 2 & 3 & \cdots & i-1 & i \\ 0 & 0 & \binom{3}{2} & \cdots & \binom{i-1}{2} & \binom{i}{2} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \binom{i-1}{i-2} & \binom{i}{i-2} \end{vmatrix}. \quad i = 1, 2, \ldots. \tag{6}$$

It is known that the Bernoulli polynomials form a complete basis over $[0, 1]$, for more details see[17]. The first few Bernoulli polynomials are

$$\mathcal{B}_0(t) = 1, \quad \mathcal{B}_1(t) = t - \frac{1}{2}, \quad \mathcal{B}_2(t) = t^2 - t + \frac{1}{6},$$
$$\mathcal{B}_3(t) = t^3 - \frac{3}{2}t^2 + \frac{1}{2}t.$$

These polynomials form a complete basis[19] over the interval $[0, 1]$ and satisfy the following equation[14]:

$$\int_0^1 \mathcal{B}_n(t)\mathcal{B}_m(t) = (-1)^{n-1} \frac{m!n!}{(m+n)!} \alpha_{n+m}, \quad m, n \geq 1. \tag{7}$$

The function $f(t)$ may be approximated by Bernoulli polynomial as follows:

$$f(t) \approx \sum_{j=0}^{m} \gamma_j \mathcal{B}_j(t) = \gamma^T \mathcal{B}(t) \tag{8}$$

where

$$\mathcal{B}(t) = [\mathcal{B}_0(t), \mathcal{B}_1(t), \cdots, \mathcal{B}_m(t)]^T, \tag{9}$$

$$\gamma = [\gamma_0, \gamma_1, \cdots, \gamma_m]^T. \tag{10}$$

such that γ uniquely calculated by

$$\gamma^T \langle \mathcal{B}(t), \mathcal{B}(t) \rangle = \langle f(t), \mathcal{B}(t) \rangle. \tag{11}$$

In equation (11), $\langle \mathcal{B}(t), \mathcal{B}(t) \rangle$ is square matrix of order $m+1$ which is defined by (7). let $D = \langle \mathcal{B}(t), \mathcal{B}(t) \rangle$, so $\gamma = D^{-1} \langle f(t), \mathcal{B}(t) \rangle$.

Lemma 1. Suppose $f(t) \in C^{m+1}[0,1]$ and $\mathcal{S}_m = \text{Span}\{\mathcal{B}_0(t), \mathcal{B}_1(t), \cdots, \mathcal{B}_m(t)\}$. If $\gamma^T \mathcal{B}(t)$ is the best approximation $f(t)$ out of $\mathcal{S}_m$, then [10]

$$||f(t) - \gamma^T \mathcal{B}(t)||_{L^2[0,1]} \leq \frac{K}{(m+1)!\sqrt{2m+3}}$$

where $K = \max_{t \in [0,1]} |f^{m+1}(t)|$.

For more details about best approximation see [19]. The exact value of the error of the estimated solution is given in the next theorem.

Theorem 1. Suppose that H is a Hilbert space and Y is a finite-dimensional closed subspace of H and $y_0, y_1, ..., y_m$ is any basis for Y. Let x be an arbitrary element in H and y_0 be the unique best approximation to x from Y. Then

$$\|x - y_0\| = \frac{G(x, y_1, y_2, \cdots, y_n)}{G(y_1, y_2, \cdots, y_n)},$$

where

$$G(x, y_1, y_2, \cdots, y_n) = \begin{vmatrix} \langle x, x \rangle & \langle x, y_1 \rangle & \cdots & \langle x, y_n \rangle \\ \langle y_1, x \rangle & \langle y_1, y_1 \rangle & \cdots & \langle y_1, y_n \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle y_n, x \rangle & \langle y_n, y_1 \rangle & \cdots & \langle y_n, y_n \rangle \end{vmatrix}.$$

2.2 | Fractional integral and derivative

There are various definitions of fractional integrations and derivatives. The FOCP in this paper in terms of the RiemannâĂŞLiouville fractional integral and the Caputo fractional derivative.

Definition 1. The Riemann-Liouville fractional integral operator of order α, is defined by

$$I^\alpha f(t) = \frac{1}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1} f(s) ds, \quad \alpha > 0, t > 0.$$

where $\Gamma(.)$ denotes the Gamma function and we set $I^0 f(t) = f(t)$.

Definition 2. Let $n = \lceil \alpha \rceil$ ($\lceil . \rceil$ denotes the ceiling function), the operator D^α, defined by

$$D^\alpha f(t) = D^n I^{n-\alpha} f(t),$$

is called the Riemann-Liouville fractional differential operator of order α. For $\alpha = 0$, we set $D^0 = I$, the identity operator.

Definition 3. The Caputo fractional derivative of $f \in L_1[0,b]$, is defined as

$$D_*^\alpha f(t) = \begin{cases} I^{n-\alpha} D^n f(t) & n-1 < \alpha < n, n \in N, \\ \frac{d^n}{dt^n} f(t) & \alpha = n. \end{cases} \tag{12}$$

Lemma 2. Let $\alpha, \beta \geq 0, C_1, C_2 \in \mathbb{R}$ and $\mathcal{F}(t), \mathcal{G}(t), \in L_1[0,1]$. Then

1) $I^\alpha I^\beta \mathcal{F}(t) = I^\beta I^\alpha \mathcal{F}(t)$,

2) $I^\alpha I^\beta \mathcal{F}(t) = I^{\beta+\alpha} \mathcal{F}(t)$,

3) The Caputo fractional differentiation is a linear operation
$D^\alpha(C_1\mathcal{F}(t) + C_2\mathcal{G}(t)) = C_1 D^\alpha \mathcal{F}(t) + C_2 D^\alpha \mathcal{G}(t)$.

4) The relation
$$I^\alpha D_*^\alpha f(t) = f(t) - \sum f^{(k)}(0^+)\frac{t^k}{k!}$$
hold almost everywhere on $[0,1]$. Note that for $n-1 < \alpha < n, n \in N$

5) $D_*^\alpha k = 0$,(k is a constant).

6)
$$D_*^\alpha t^\nu = \begin{cases} 0 & for \nu \in N_0 and \nu < \lceil \alpha \rceil \\ \frac{\Gamma(\nu+1)}{\Gamma(\nu+1--\alpha)} t^{\nu-\alpha} & for \nu \in N_0 and \nu \geq \lceil \alpha \rceil \\ & or \nu \in N and \nu > \lceil \alpha \rceil \end{cases} \tag{13}$$

Bernoulli operational matrix of the fractional integration The Rimann-Lioville fractional integration of the vector $\mathcal{B}(t)$ in equation (9) can be expressed by

$$I^\alpha \mathcal{B}(t) = F^{(\alpha)} \mathcal{B}(t) \tag{14}$$

where the square matrix $F^{(\alpha)}$ of order $m+1$ is Riemann-Liouville fractional operational matrix of integration. This matrix is given in [18] as follows:

$$F^\alpha = \begin{bmatrix} \theta_{0,0,0} & \theta_{0,1,0} & \cdots & \theta_{0,m,0} \\ \sum_{r=0}^{1} \theta & \sum_{r=0}^{1} \theta & \cdots & \sum_{r=0}^{1} \theta \\ \vdots & \vdots & \cdots & \vdots \\ \sum_{r=0}^{m} \theta & \sum_{r=0}^{m} \theta & \cdots & \sum_{r=0}^{m} \theta \end{bmatrix} \tag{15}$$

where

$$\theta_{i,j,r} = b_{i,r}^{(\alpha)} c_{r,j}. \tag{16}$$

In equation (16), $b_{i,r}^{(\alpha)}$ and $c_{r,j}$ is given by

$$b_{i,r}^{(\alpha)} = \frac{i!}{(i-r)!\Gamma(r+1+\alpha)} \alpha_{i-r}.$$

and $c_{r,j}$ are the coefficient expansion of function t^{q+r} in terms of Bernoulli polynomials

$$t^{\alpha+r} \approx \sum_{j=0}^{m} c_{r,j} \mathcal{B}_j(t).$$

This matrix for $m = 5$ and $\alpha = 0.8, 0.9, 1$ are given as follows

$$F^{(0.8)} = \begin{pmatrix} 0.596484 & 1.06274 & -0.422194 & 0.930355 & -0.535861 & 1.10955 \\ -0.085212 & 0.0653276 & 0.734063 & -0.590821 & 0.332406 & -0.63479 \\ -0.00149495 & 0.00646345 & 0.00458001 & 0.627878 & -0.19236 & 0.288085 \\ 0.00840908 & -0.00433996 & 0.00314778 & 0.0147215 & 0.339961 & -0.0971617 \\ 0.000414977 & -0.00182432 & 0.00833749 & -0.0293234 & 0.0914694 & 0.230468 \\ -0.00399701 & 0.00206342 & -0.0132906 & 0.0130937 & -0.211251 & 0.0900255 \end{pmatrix}$$

$$F^{(0.9)} = \begin{pmatrix} 0.547239 & 1.03612 & -0.193272 & 0.377605 & -0.219211 & 0.433338 \\ -0.0849164 & 0.0292591 & 0.611414 & -0.247547 & 0.138349 & -0.250288 \\ -0.000725781 & 0.00276464 & 0.000839296 & 0.464273 & -0.0844637 & 0.116276 \\ 0.00844276 & -0.00212171 & 0.00150434 & 0.00473369 & 0.295352 & -0.0436683 \\ 0.000202395 & -0.000781233 & 0.00352227 & -0.011622 & 0.0385064 & 0.219963 \\ -0.0040171 & 0.00100958 & -0.010285 & 0.00570285 & -0.181167 & 0.0382047 \end{pmatrix}$$

$$F^{(1)} = \begin{pmatrix} \frac{1}{2} & 1 & 0 & 0 & 0 & 0 \\ -\frac{1}{12} & 0 & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{3} & 0 & 0 \\ \frac{1}{120} & 0 & 0 & 0 & \frac{1}{4} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{5} \\ -\frac{1}{252} & 0 & -\frac{1}{132} & 0 & -\frac{5}{33} & 0 \end{pmatrix}$$

In[10], an error upper bound for the operational matrix of the fractional integration F^{α} has given and demonstrate with an increase in the number of Bernoulli polynomials, the error vector given in theorem 2 tend to zero vector. Here, we restate this theorem.

Theorem 2. Suppose $f(t) \in L^2[0,1]$ is approximated by $\gamma^T \mathcal{B}(t)$ where $\mathcal{B}(t)$ and γ are defined in equations (9) and (10) then we have

$$lim_{m\to\infty} ||f(t) - \gamma^T \mathcal{B}(t)||_{L^2[0,1]} = 0$$

The error vector $e_I^q = [e_{I0}^q, e_{I1}^q, \cdots, e_{Im}^q]^T = F^{(q)}\mathcal{B}(t) - I^q\mathcal{B}(t)$, Then[10]

$$||e_{Ii}^q|| \leq \sum_{r=0}^{i} \frac{i!}{(i-r)!\Gamma(r+1+q)} \alpha_{i-r} \left(\frac{G(t^{q+r}, \mathcal{B}_0(t), \cdots, \mathcal{B}_m(t))}{G(\mathcal{B}_0(t), \cdots, \mathcal{B}_m(t)))} \right)^{\frac{1}{2}}, \quad 0 \leq i \leq m. \tag{17}$$

where $G(t^{q+r}, \mathcal{B}_0(t), \cdots, \mathcal{B}_m(t))$ and $G(\mathcal{B}_0(t), \cdots, \mathcal{B}_m(t))$ is defined in theorem 1.

3 | SOLUTION FRAMEWORK

3.1 | Discretization of FOCP

In this section, implementation of the Bernoulli polynomials on FOCP in equations (1)-(3) is given. Firstly, the control and fractional state rate functions are approximated in terms of Bernoulli polynomials of the form

$$D^{\alpha}x(t) = \mathcal{X}^T \mathcal{B}(t), \tag{18}$$

$$u(t) = \mathcal{U}^T \mathcal{B}(t) \tag{19}$$

where

$$\mathcal{X}^T = [\mathcal{X}_0, \mathcal{X}_1, \cdots, \mathcal{X}_m], \qquad \mathcal{U}^T = [\mathcal{U}_0, \mathcal{U}_1, \cdots, \mathcal{U}_m]. \tag{20}$$

are unknown and $\mathcal{B}(t)$ is given in (9). The state functions $x(t)$ can be obtained by using equations (13) and (14)

$$x(t) = I^{\alpha}D^{\alpha}x(t) + x(0) \approx (\mathcal{X}^T F^{(\alpha)} + d^T)\mathcal{B}(t), \tag{21}$$

where F^q is fractional operational matrix of integration introduced in (15) and the constant vector d is $d^T = [x_0, \cdots, 0]$. By using equations (18), (19) and (21) in dynamical system (2), we obtain

$$\mathcal{Q}(t) = \mathcal{G}\left(t, (\mathcal{X}^T F^{(\alpha)} + d^T)\mathcal{B}(t)\right) + \beta(t)(U^T\mathcal{B}(t)) - \left(\mathcal{X}^T\mathcal{B}(t)\right) = 0 \tag{22}$$

Now, we utilize collocation points $\tau_{i+1} = \frac{1}{2}\left(Cos\left(\frac{(i\Pi)}{m+1}\right) + 1\right)$ for $i = 0, \cdots, m$ in (22), and (22) is converted to

$$\mathcal{Q}(\tau_{i+1}) = 0, \quad i = 0, \cdots, m. \tag{23}$$

For approximating the cost function stated in (1), we apply the GL quadrature after the appropriate interval transformation and use (19) and (21), so we obtain

$$\mathcal{J}(t) = \int_0^1 \mathcal{F}(t, (\mathcal{X}^T F^{(\alpha)} + d^T)\mathcal{B}(t), \mathcal{U}^T\mathcal{B}(t))dt$$

$$= \frac{1}{2}\int_{-1}^{1} \mathcal{F}\left(\frac{\delta+1}{2}), (\mathcal{X}^T F^{(\alpha)} + d^T)\mathcal{B}(\frac{\delta+1}{2}), \mathcal{U}^T\mathcal{B}(\frac{\delta+1}{2})\right) d\delta$$

$$\approx \frac{1}{2}\sum_{i=0}^{m}\omega_i \mathcal{F}\left(\frac{\delta_i+1}{2}, (\mathcal{X}^T F^{(\alpha)} + d^T)\mathcal{B}(\frac{\delta_i+1}{2}), \mathcal{U}^T\mathcal{B}(\frac{\delta_i+1}{2})\right) \tag{24}$$

where δ_i, $i = 0, \cdots m$ are $m+1$ zeros of Legendre polynomials and ω_i are the corresponding weights[5]. The initial condition (3) can be written as

$$x(0) = x_0 = (\mathcal{X}^T F^{(\alpha)} + d^T)\mathcal{B}(0). \tag{25}$$

Then, the FOCP in equations (1)-(3) has been reduced to a NLP with (24) as the objective function and (23) and (25) as constraints. The next subsections are devoted to the description of MVO, MFO and WOA algorithms, to solve NLP governed by FOCP.

3.2 | Multi-Verse Optimizer

MVO is a novel population-based algorithm which was proposed by Mirjalili et al.[21]. This algorithm is virtually extended based on the concepts of white hole and black hole. In MVO, the search space exploration is according to the fundamental principles of the white hole and black hole. Also, the wormholes help MVO in exploiting the search areas. Each solution was similar to a universe and each variable was an item in that universe. In addition, an inflation rate was assigned to each solution, which was relative to the fitness function value.

A roulette wheel strategy was executed to model the white/black hole tunnels and swap the objects of universes. The universes were sorted regarding their inflation rates and selected based on the roulette wheel for a white hole on every iteration. The following steps were performed to carry out this.

$$U = \begin{bmatrix} x_1^1 & x_1^2 & \dots & x_1^d \\ x_2^1 & x_2^2 & \dots & x_2^d \\ \vdots & \vdots & \ddots & \vdots \\ x_n^1 & x_n^2 & \dots & x_n^d \end{bmatrix}$$

where d is the number of variables and n is the number of universes or candidate solutions:

$$x_i^j = \begin{cases} x_k^j, & r1 < NI(U_i) \\ x_i^j, & r1 \geq NI(U_i) \end{cases} \tag{26}$$

where x_i^j demonstrated the jth parameter of ith universe, U_i indicated the ith universe, $NI(Ui)$ was normalized inflation rate of the ith universe, $r1$ was a random number in the interval $[0, 1]$, and x_k^j showed the jth parameter of kth universe chosen by a roulette wheel selection strategy.

The pseudocodes for this step were as follows:

It can be seen in Eq. (26), the selection of white holes were performed by the roulette wheel, which was with respect

```
SU=Sorted universes
NI=Normalize inflation rate (fitness) of the universes
for each universe indexed by i do
    Black hole index=i;
    for each object indexed by j do
        r1 = random([0,1]);
        if r1 < NI(U_i) then
            White hole index= Roulette Wheel Selection(-NI);
            U(Black hole index,j)= SU(White hole index,j);
        end if
    end for
end for
```

to the normalized inflation rate. The low inflation rate, the more probability of transferring objects by white/black

hole tunnels. Each universe had wormholes to send its objects over area randomly to keep the diversity of universes and execute exploitation. The wormhole tunnels were always established between a universe and the best universe obtained hitherto to ensure local changes for each universe and gain high probability of making better inflation rate concerning wormholes. The mathematical model of this strategy was as follows:

$$x_i^j = \begin{cases} \begin{cases} x_j + TDR \times ((ub_j - lb_j) \times r4 + lb_j), & r3 < 0.5 \\ x_j - TDR \times ((ub_j - lb_j) \times r4 + lb_j), & r3 \geq 0.5 \end{cases}, & r2 < WEP \\ x_i^j, & r2 \geq WEP \end{cases} \tag{27}$$

where x_j demonstrated the jth parameter of best universe obtained hitherto, wormhole existence probability (WEP) and traveling distance rate (TDR) were two coefficients, lb_j indicated the lower bound of jth variable, ub_j was the upper bound of jth variable, x_i^j showed the jth parameter of ith universe, and $r2, r3, r4$ were random numbers in the interval $[0, 1]$. The pseudo codes were as follows (on the assumption that ub and lb showed upper bound and lower bound of the decision variables):

The adaptive formula for WEP and TDR coefficients were as follows:

```
for each universe indexed by i do
    for each object indexed by j do
        r2 = random([0,1]);
        if r2 < Wormhole existance probability then
            r3 = random([0,1]);
            r4 = random([0,1]);
            if r3 < 0.5 then
                U(i,j)=Best universe(j) + Travelling distance rate *((ub(j) - lb(j)) * r4 + lb(j));
            else
                U(i,j)=Best universe(j) - Travelling distance rate *((ub(j) - lb(j)) * r4 + lb(j));
            end if
        end if
    end for
end for
```

$$WEP = min + l \times (\frac{max - min}{L}) \tag{28}$$

where min was 0.2, max was 1, l demonstrated the current iteration, and L indicated the maximum iterations.

$$TDR = 1 - \frac{l^{\frac{1}{p}}}{L^{\frac{1}{p}}} \tag{29}$$

where p represented the exploitation precision over the iterations. The p was 6 in this study.

The steps of the MVO algorithm were specified as follows:

3.3 | Moth-Flame Optimization

The MFO was a recently population based algorithm belonging to the class of nature-inspired approaches. It was proposed and developed by Mirjalili [22]. In the MFO algorithm, the candidate solutions were moths and the variables were the position of moths in the area. So, the moths could fly with varying their position vectors. The set of moths was demonstrated in a matrix as follows:

$$M = \begin{bmatrix} m_{1,1} & m_{1,2} & \dots & m_{1,d} \\ m_{2,1} & m_{2,2} & \dots & m_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n,1} & m_{n,2} & \dots & m_{n,d} \end{bmatrix} \tag{30}$$

where n was the number of moths and d was the number of variables.

Algorithm 1 MVO algorithm

Generate random universes (U)
Initialize WER, TDR, and Best universe
SU=Sorted universes
NI=Normalize the inflation rate (fitnesses) of the universes
while the stopping criterion is not fulfilled **do**
    Calculate the fitness of all universes
    for each universe indexed by i **do**
        Update WEP and TDR
        Black hole index=i;
        for each object indexed by j **do**
            $r1 = random([0,1])$;
            if $r1 < NI(U_i)$ **then**
                White hole index= Roulette Wheel Selection(-NI);
                U(Black hole index,j)=SU(White hole index,j);
            end if
            $r2 = random([0,1])$;
            if $r2 <$ Wormhole existence probability **then**
                $r3 = random([0,1])$;
                $r4 = random([0,1])$;
                if $r3 < 0.5$ **then**
                    $U(i,j)$=Best universe(j) + Traveling distance rate $*((ub(j) - lb(j)) * r4 + lb(j))$;
                else
                    $U(i,j)$=Best universe(j) - Travelling distance rate $*((ub(j) - lb(j)) * r4 + lb(j))$;
                end if
            end if
        end for
    end for
end while

Also, an array for computing the fitness values of moths was considered:

$$OM = \begin{bmatrix} OM_1 \\ OM_2 \\ \vdots \\ OM_n \end{bmatrix} \tag{31}$$

where n was the number of moths. On the other hand, a matrix was assigned to flames as follows:

$$F = \begin{bmatrix} F_{1,1} & F_{1,2} & \dots & F_{1,d} \\ F_{2,1} & F_{2,2} & \dots & F_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ F_{n,1} & F_{n,2} & \dots & F_{n,d} \end{bmatrix} \tag{32}$$

where n was the number of moths and d was the number of variables. Also, there was an array for computing the fitness values of flames as follows:

$$OF = \begin{bmatrix} OF_1 \\ OF_2 \\ \vdots \\ OF_n \end{bmatrix} \tag{33}$$

where n was the number of moths. The moths were search agents that travel around the search area, while flames were the best position of moths that achieved hitherto. So, each moth explored around a flame and updated it for detecting a better solution. The MFO algorithm was a three-tuple that described as follows:

$$MFO = (I, P, T) \tag{34}$$

I was a function of moths and fitness values that was defined as follows:

$$I : \phi \rightarrow \{M, OM\} \tag{35}$$

The P was the main function, traveled the moths around the search area. This function gained the matrix of M and gone back its updated.

$$P : M \rightarrow M \tag{36}$$

The T function considered true if the stopping criterion was fulfilled and false if the stopping criterion is not fulfilled:

$$T : M \rightarrow \{true, false\} \tag{37}$$

With $I, P, andT$, the MFO algorithm was described as follows: The function I had to create initial solutions and

```
M=I();
while T(M) is equal to false do
    M = P(M);
end while
```

compute the objective functions. The following mechanism was applied as a random distribution in this function:

```
for i = 1 : n do
    for j = 1 : d do
        M(i, j) = (ub(i) − lb(i)) ∗ rand() + lb(i);
    end for
end for
OM = FitnessFunction(M);
```

there are two other arrays called ub and lb. Two matrixes ub and lb given the upper and lower bounds of the variables and were computed by:

$$ub = [ub_1, ub_2, ..., ub_{n-1}, ub_n] \tag{38}$$

$$lb = [lb_1, lb_2, ..., lb_{n-1}, lb_n] \tag{39}$$

Then, the P function was iteratively performed until the T function reflected true. The P function was the major function that traveled the moths around the search area. Therefore, the situation of each moth was defined based on the following equation:

$$M_i = S(M_i, F_j) \tag{40}$$

where M_i demonstrated the i-th moth, F_j showed the j-th flame, and S was the spiral function.

A logarithmic spiral was described as the main update method of moths as follows:

$$S(M_i, F_j) = D_i.e^{bt}.cos(2\pi t) + F_j \tag{41}$$

where D_i showed the distance of the i-th moth for the j-th flame, b was a fixed value, and t was a random number in the interval $[-1, 1]$. D was computed by:

$$D_i = |F_j - M_i| \tag{42}$$

where M_i demonstrated the i-th moth, F_j showed the j-th flame, and D_i displayed the distance of the i-th moth for the j-th flame.

Eq. (41) was in which the spiral flying path of moths was mimicked. To highlight exploitation, t was considered as a random number in the interval $[r, 1]$ in which r was linearly reduced from -1 to -2 on the iterations. The following formula was used:

$$\text{flame no} = round(N - 1 * \frac{N-1}{T}) \tag{43}$$

The steps of the P function were as below. The P function was performed until the T function come back true.

```
Update flame no using Eq. (43)
OM = FitnessFunction(M);
if iteration == 1 then
    F = sort(M);
    OF = sort(OM);
else
    F = sort(M_{t-1}, M_t);
    OF =sort(M_{t-1}, M_t);
end if
for i = 1 : n do
    for j = 1 : d do
        Update r and t
        Compute D using Eq. (42) based on the corresponding moth
        Update M(i, j) using Eqs. (40) and (41) based on the corresponding moth
    end for
end for
```

Then, the best moth was considered as the best achieved approximation of the minimum.

3.4 | Whale Optimization Algorithm

WOA was a new nature-inspired metaheuristic approach which was introduced by Mirjalili and Lewis [23] in the 2016s. The encircling prey, spiral bubble-net feeding maneuver, and search for prey were three important elements of the WOA algorithm. A brief description of related elements on WOA was discussed in the following.

3.4.1 | Encircling prey

Humpback whales can identify the position of prey and surround them. Considering the location of the optimal prey in the search region was not known, the present best candidate solution was the optimal prey or was close to the minimum in the WOA. Subsequently the best search agent was determined, the other search agents would attempt to update their locations in the direction of the best search agent. This action was shown by the following equations:

$$\overrightarrow{D} = |\overrightarrow{C}.\overrightarrow{X^*}(t) - \overrightarrow{X}(t)| \tag{44}$$

$$\overrightarrow{X}(t+1) = \overrightarrow{X^*}(t) - \overrightarrow{A}.\overrightarrow{D} \tag{45}$$

where t demonstrated the current iteration, $\overrightarrow{A}$ and $\overrightarrow{C}$ were coefficient vectors, X^* was the location vector of the best solution achieved hitherto, and $\overrightarrow{X}$ was the location vector. The vectors $\overrightarrow{A}$ and $\overrightarrow{C}$ were computed as follows:

$$\overrightarrow{A} = 2\overrightarrow{a}.\overrightarrow{r} - \overrightarrow{a} \tag{46}$$

$$\overrightarrow{C} = 2.\overrightarrow{r} \tag{47}$$

where $\overrightarrow{a}$ was linearly reduced from 2 to 0 on the iterations and $\overrightarrow{r}$ was a random vector in the interval $[0, 1]$.

3.4.2 | Bubble-net attacking method

Two plans were considered to model the bubble-net behavior of humpback whales as follows:

1. *Shrinking surrounding structure*: This action was obtained by reducing the value of $\overrightarrow{a}$ in the Eq. (46). Also, the fluctuation range of $\overrightarrow{A}$ was reduced by $\overrightarrow{a}$. Especially $\overrightarrow{A}$ was a random value in $[a, a]$ in which a was reduced from 2 to 0 on the iterations. Considering random values for $\overrightarrow{A}$ in $[1, 1]$, the new location of a search agent can be determined between the original location of the agent and the location of the current best agent.

2. *Spiral updating location*: This action first computed the distance between the whale settled at (X, Y) and prey settled at (X^*, Y^*). Then, a spiral equation was generated between the location of whale and prey to simulate the helix-shaped movement of humpback whales by:

$$\overrightarrow{X}(t+1) = \overrightarrow{D'}.e^{bl}.cos(2\pi l) + \overrightarrow{X^*}(t) \tag{48}$$

where $\overrightarrow{D'} = |\overrightarrow{X^*}(t) - \overrightarrow{X}(t)|$ and demonstrated the distance of the ith whale to the prey, b was a fixed value for describing the shape of the logarithmic spiral, and l was a random number in the interval $[1, 1]$.

The humpback whales moved around the prey inside a shrinking ring and near a spiral-shaped path. On the other hand, there was a probability of 50% to select between either the shrinking surrounding method or the spiral approach to update the location of whales. This mechanism was formulated as follows:

$$\overrightarrow{X}(t+1) = \begin{cases} \overrightarrow{X^*}(t) - \overrightarrow{A}.\overrightarrow{D}, & p < 0.5 \\ \overrightarrow{D'}.e^{bl}.cos(2\pi l) + \overrightarrow{X^*}(t), & p \geq 0.5 \end{cases} \tag{49}$$

where p was a random number in the interval $[0, 1]$.

3.4.3 | Search for prey

In this mechanism, humpback whales searched randomly with respect to the location of each other. So, $\overrightarrow{A}$ was applied with the random values larger than 1 or smaller than 1 to restrict search agent to travel far away from a particular whale. This method and $|\overrightarrow{A}| > 1$ highlight exploration in the WOA algorithm to execute a global search as follows:

$$\overrightarrow{D} = |\overrightarrow{C}.\overrightarrow{X_{rand}} - \overrightarrow{X}| \tag{50}$$

$$\overrightarrow{X}(t+1) = \overrightarrow{X_{rand}} - \overrightarrow{A}.\overrightarrow{D} \tag{51}$$

where $\overrightarrow{X_{rand}}$ was a random whale selected from the present population.

The pseudo code of the WOA was stated by:

4 | UTILIZATION OF THE PRESENTED APPROACH

In order to demonstrate and confirm the performance and the precision of our scheme, we propounded three following examples. *Matlab* software had been utilized to dissolve these illustrative examples. After the discritization of main problem with proposed method in subsection 3.1, the resulted NLP has been solved with three powerful optimization methods MFO, MVO and WOA. We calculate the absolute error of the approximate optimal state and control variables as follows

$$E_X(\tau) = |x^*(\tau) - x(\tau)|, \quad E_u(\tau) = |u^*(\tau) - u(\tau)|. \tag{52}$$

Algorithm 2 WOA algorithm

Initialize the whales population $X_i(i = 1, 2, ..., n)$
Evaluate the fitness of each search agent
X^*=the best search agent
while ($t <$ maximum number of iterations) **do**
  for each search agent **do**
    Update a, A, C, l, and p
    if 1 ($p < 0.5$) **then**
      if 2 ($|A| < 1$) **then**
        Update the location of the present search agent by the Eq. (44)
      else2 ($|A| \geq 1$)
        Choose a random search agent (X_{rand})
        Update the location of the present search agent by the Eq. (51)
      end if2
    else1 ($p \geq 0.5$)
      Update the location of the present search by the Eq. (48)
    end if1
  end for
  Check if any search agent goes beyond the search region and amend it
  Evaluate the fitness of each search agent
  Update X^* if there is a better solution
  $t = t + 1$
end while
return X^*

4.1 | Numerical examples

Example 1. Consider the following optimal control problem

$$\min \quad \mathcal{J} = \frac{1}{2}\int x^2(t) + u^2(t)dt \tag{53}$$

contingent to fractional system dynamics

$$D^\alpha x(t) = -x(t) + u(t), \quad x(0) = 1. \tag{54}$$

The exact solution for $\alpha = 1$ is

$$x(t) = \cosh(\sqrt{2t}) + \eta \sinh(\sqrt{2t}),$$
$$u(t) = (1 + \sqrt{2\eta})\cosh(\sqrt{2t}) + (\sqrt{2} + \eta)sinh(\sqrt{2t}).$$

where

$$\eta = -\frac{\cosh(\sqrt{2}) + \sqrt{2}\sinh(\sqrt{2})}{\sqrt{2}\cosh(\sqrt{2}) + \sinh(\sqrt{2})}.$$

The maximum absolute error of our scheme for state and control variables for $M = 5$ and $\alpha = 1$ are plotted in figures 1 and 2 . Table 1 demonstrates the value of objective functional obtained for $M = 5, 6$ and different values of α. The numerical solutions of this FOCP are also comparable with other methods in the literature, for this purpose we refer to[18]. It is found that the results obtained by the proposed method shows very good agreement with published results in[18]. Moreover, our proposed method has obtained 7th order of accuracy.

Example 2. Consider the objective functional and state space equations as follows:

$$\min J = \int_0^1 \Big(x^2(t) - 2t^{\frac{3}{2}}x(t) + u^2(t) - \frac{3\sqrt{\pi}}{4}e^{-t}u(t) + e^{-t+t^{\frac{3}{2}}}u(t) + t^3 + \frac{9\pi}{64}e^{-2t}$$
$$-\frac{3\sqrt{\pi}}{8}e^{-2t+t^{\frac{3}{2}}} + \frac{1}{4}e^{-2t+t^{\frac{3}{2}}} + e^{2t}\Big)dt \tag{55}$$

TABLE 1 The value of objective functional for example 1

M	α	**MFO**	**MVO**	**WOA**
5	0.8	0.0607742130726983	0.0607749705120333	0.0607752304090174
5	0.9	0.0564543252057422	0.0564758737200734	0.0570053588731675
5	1.0	0.0539772578587941	0.0547324421838168	0.0541010362352063
6	0.8	0.0604090556427560	0.0604093005209958	0.0604097171588314
6	0.9	0.0553258723321114	0.0553540556629651	0.0553309922852756
6	1.0	0.0527690492188603	0.0533263387871045	0.0529847532371575

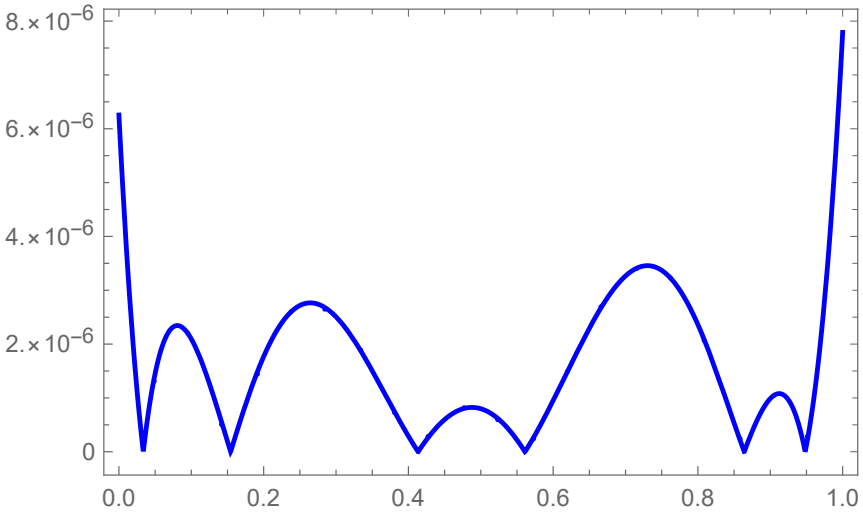

FIGURE 1 The maximum absolute error $E_x(t)$ in example 1

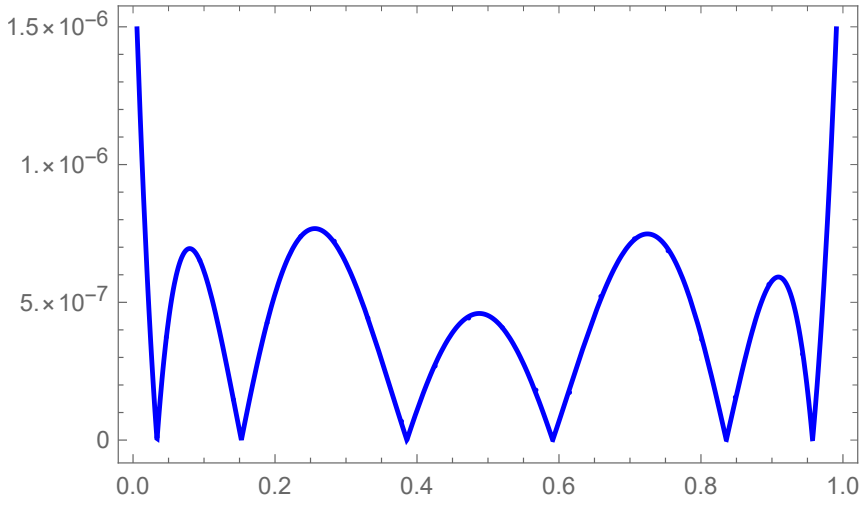

FIGURE 2 The maximum absolute error $E_u(t)$ in example 1

subject to

$$D^{1.5}x(t) = e^{x(t)} + 2e^t u(t),$$

$$x(0) = x'(0) = 0. \tag{56}$$

The optimal control and state functions for this example are $u^*(t) = \frac{1}{2}e^{-t}\left(-e^{t^{\frac{3}{2}}} + \frac{3\sqrt{\pi}}{4}\right)$ and $x^*(t) = t^{\frac{3}{2}}$. By utilizing the procedure given in section 3, firstly the FOCP is discretized using the Bernoulli polynomials and its operational matrix o and then the resulted NLP is optimized using the three given approaches MFO, MVO and WOA. The value of cost functions for $M = 2, 3$ and 4 by using three optimization methods are given in Table 2 . The exact and approximate state and control functions are plotted in Figures 3 and 4 . In these figures, the blue graph represents the exact function and graph with red dots is approximate functions. For $M = 4$, the operational matrix of fractional integration has been obtained as follows

$$F^{(1.5)} = \begin{pmatrix} 0.300901 & 0.0644788 & 0.0023881 & -0.00651301 & -0.000678021 \\ -0.0644788 & -0.00835836 & 0.000976952 & 0.000918502 & -0.00028223 \\ 0.0023881 & -0.000976952 & -0.000214317 & 0.00013026 & 0.0000644752 \\ 0.00651301 & 0.000918502 & -0.00013026 & -0.000108059 & 0.0000388495 \\ -0.000678021 & 0.00028223 & 0.0000644752 & -0.0000388495 & -0.0000196873 \end{pmatrix}$$

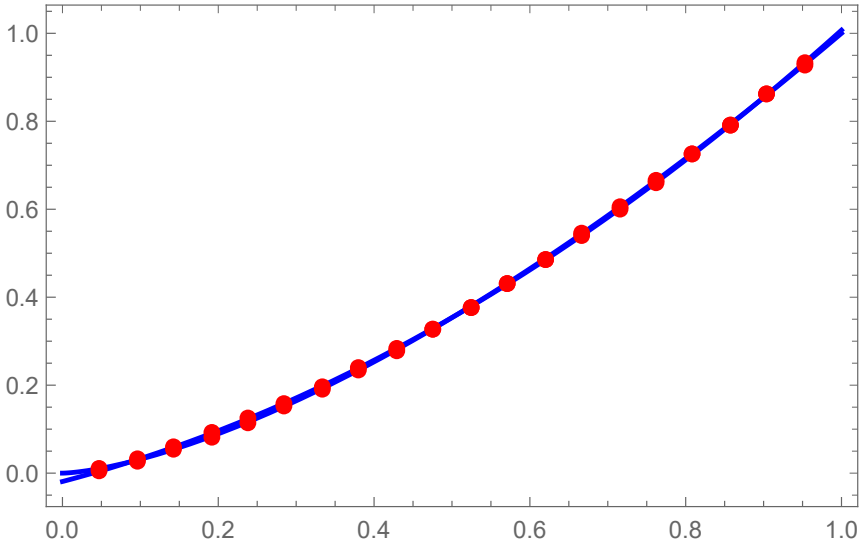

FIGURE 3 Exact and approximate state function for example 2

TABLE 2 The value of objective functional for example 2

M	MFO	MVO	WOA
2	3.19454812683797	3.19454854799758	3.19454849446344
3	3.06165629681816	3.06165629836741	3.06165630045092
4	1.12915953267903	1.12915953355528	1.12915957556041

Example 3. Consider the following FOCP

$$\min J = \int_0^1 \Big(-2e^{1+t^2+x(t)} + e^{2(1+t^2+x(t))} + \frac{8\sqrt{t}}{\sqrt{\pi}}u(t) - 2\sin(1+t^2)u(t) + u^2(t)$$

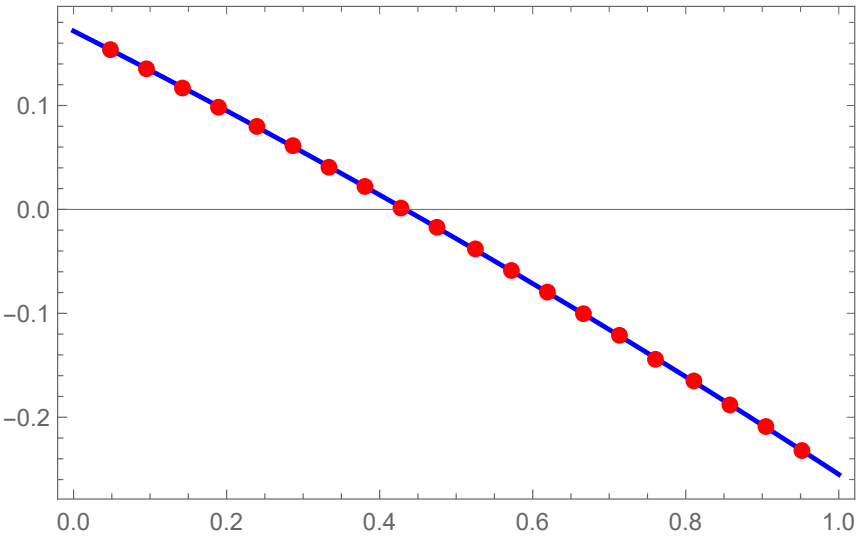

FIGURE 4 Exact and approximate control function for example 2

$$\frac{16t}{\pi} - \frac{8\sqrt{tsin(1+t^2)}}{\sqrt{\pi}+\sin^2(1+t^2)+1}\Big)dt$$

subject to

$$D^{1.5}x(t) = \sin x(t) + u(t), \quad x(0) = -1, \quad x'(0) = 0.$$

For this problem, we have $x^*(t) = -1-t^2$ and $u^*(t) = \frac{-4\sqrt{t}}{\sqrt{\pi}} + \sin(1+t^2)$. The values of objective functional for $M = 3$ and $M = 4$ are given in Table 3 . The exact and approximate state and control functions are plotted in Figures 5 and 6 . In these figures, the blue graph represents the exact function and graph with red dots is approximate functions.

TABLE 3 The value of objective functional for example 3

M	**MFO**	**MVO**	**WOA**
3	$8.67774875395422E{-}05$	$8.72990185631438E{-}05$	$8.70333367712425E{-}05$
4	$3.90521852385846E{-}05$	$3.92991862076560E{-}05$	$3.92124829314838E{-}05$

4.2 | Discussion and convergence speed

Tables 1 , 2 and 3 provide the results of executing MFO, MVO and WOA algorithms on solving the controlled fractional differential equations.Numerical results confirm the satisfactory performance of the MFO algorithm on solving three illustrative examples. It is observed from Tables 1 , 2 and 3 that the MFO is much better than MVO and WOA methods in precision that these results achieved the value of objective functional for different values of M. This implies that the MVO and WOA obtain competitive solutions in comparison to the MFO algorithm. But, the MFO algorithm outperforms the MVO and WOA methods and gets the best solutions on solving the controlled fractional differential equations. On the other hand, the quality of the numerical solutions in Figs. 1 ,2 ,3 ,4 ,5 and 6 demonstrates that the MFO favourably converges to the exact solutions. The convergence behavior of the MFO, MVO and WOA algorithms on Example 1 for $M = 6$, Example 2 for $M = 2$ and Example 3 for $M = 4$ are given in Figs. 7 , 8 and 9 , respectively. The curves demonstrate that the MFO algorithm considerably becomes better its performance during iterations. As the number of iterations ascends, the value of objective functional substantially reduces. As a result, the MFO has the capability to solve the FOCP successfully. The results indicate that MFO has

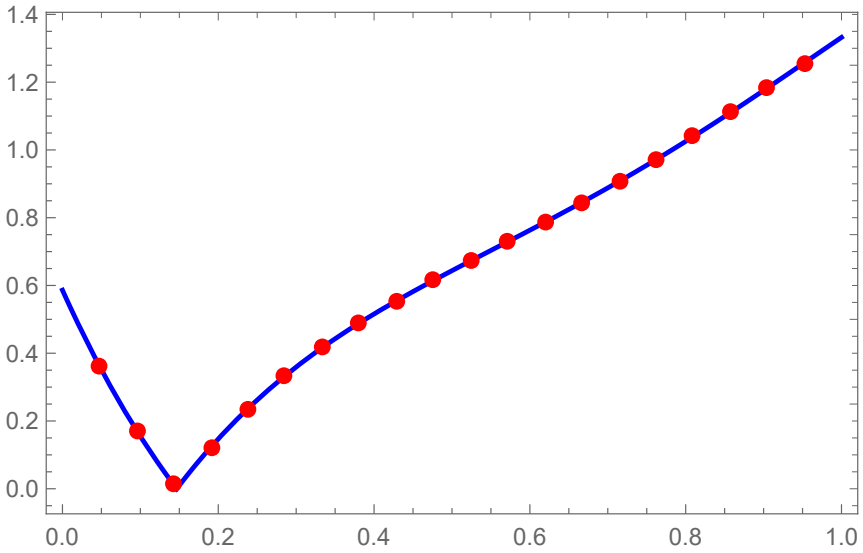

FIGURE 5 Exact and approximate control function for example 3

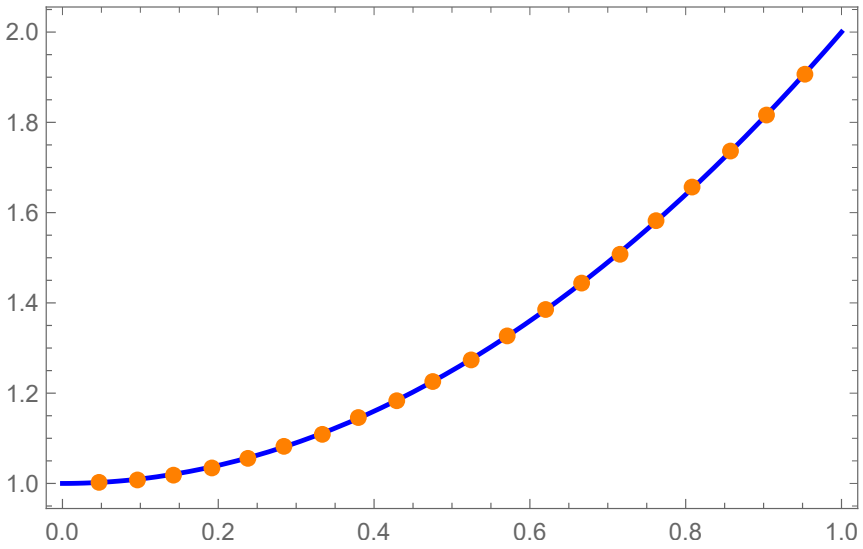

FIGURE 6 Exact and approximate control function for example 3

the quick convergence speed and MVO and WOA BSO algorithms requires more evaluation functions to converge to the exact solution. Lastly, the MVO algorithm gets the third degree in converging to the exact solution.

5 | CONCLUSIONS

In this essay, a direct method has been propounded to solve a class of fractional optimal control problem (FOCP). In this approach, the FOCP has been transmitted to a nonlinear programming problem (NLP) by using Bernoulli polynomials and operational matrix of fractional integration. Then, the performances of three nature-inspired meta-heuristics called multi-verse optimizer (MVO), moth-flame optimization (MFO), and whale optimization algorithm

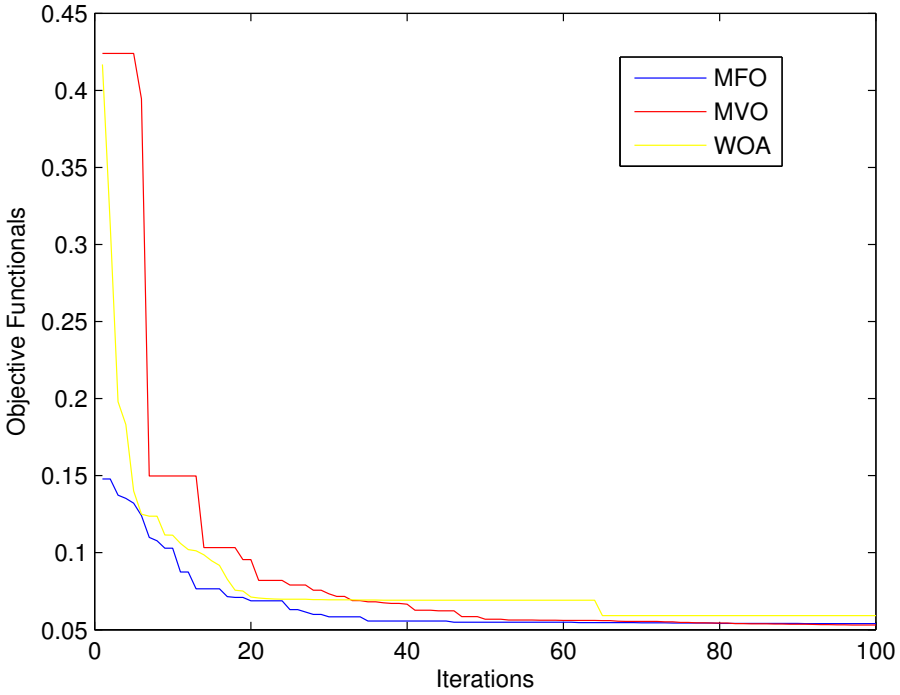

FIGURE 7 Convergence behaviors of the MFO, MVO and WOA algorithms when solving Example 1.

(WOA) have been investigated on three illustrative examples. The comparative study demonstrated that MFO represented promising solutions and surpass the MVO and WOA algorithms over three numerical examples. In addition, more efficiency is provided by incorporating the new discretization technique into the metaheuristic algorithms in solving FOCP. Also, numerical results show the simplicity and efficiency of the proposed method to dissolve the FOCP. The first author would like to appreciate the research council of Farhangian University for supporting this research. The second author would like to thank Gonbad Kavous University for supporting this research work. The work also has been supported by research council of Young Researchers and Elite Club, North Tehran Branch, Islamic Azad University, Tehran for the third author.

References

1. Agrawal, O. A General Formulation and Solution Scheme for Fractional Optimal Control Problems. Nonlinear Dyn 38, 323–337 (2004). https://doi.org/10.1007/s11071-004-3764-6

2. Bhrawy, A.H., Ezz-Eldien, S.S. A new Legendre operational technique for delay fractional optimal control problems. Calcolo 53, 521–543 (2016). https://doi.org/10.1007/s10092-015-0160-1

3. Agrawal, O. P., Defterli, O. and Baleanu, D. (2010). Fractional Optimal Control Problems with Several State and Control Variables. Journal of Vibration and Control, 16(13), 1967-âĂŞ1976. https://doi.org/10.1177/1077546309353361

4. Abubakar Bello Salati, Mostafa Shamsi, DelïňĄm F. M. Torres, Direct transcription methods based on fractional integral approximation formulas for solving nonlinear fractional optimal control problems,Communications in Nonlinear Science and Numerical Simulation(2018),doi: 10.1016/j.cnsns.2018.05.011

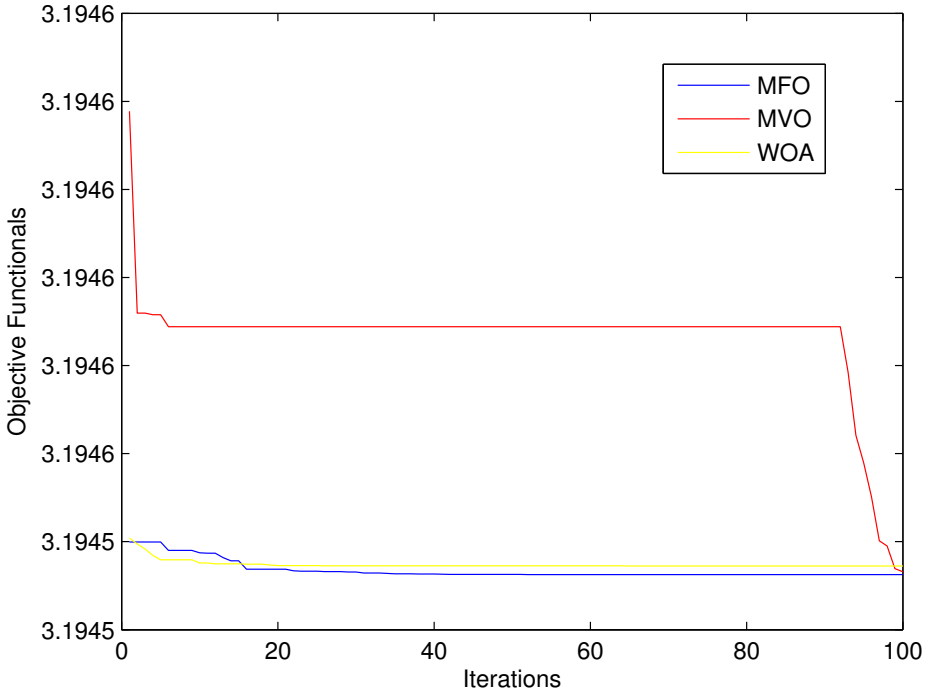

FIGURE 8 Convergence behaviors of the MFO, MVO and WOA algorithms when solving Example 2.

5. Devore RA and Scott LR (1984) Error bounds for Gaussian quadrature and weighted polynomial approximation. SIAM Journal on Numerical Analysis 21: 400–412.

6. Rivlin TJ (1981) An Introduction to the Approximation of Functions. New York: Dover Publications.

7. Zeid SS, Effati S, Kamyad AV (2018) Approximation methods for solving fractional optimal control problems. Comput Appl Math 37(1):158–182

8. Agrawal, O. P. (2008). A Formulation and Numerical Scheme for Fractional Optimal Control Problems. Journal of Vibration and Control, 14(9–10), 1291–1299. https://doi.org/10.1177/1077546307087451

9. Agrawal, O. P. and Baleanu, D. (2007). A Hamiltonian Formulation and a Direct Numerical Scheme for Fractional Optimal Control Problems. Journal of Vibration and Control, 13(9âĂŞ10), 1269–1281. https://doi.org/10.1177/1077546307077467

10. Keshavarz, E., Ordokhani, Y., and Razzaghi, M. (2016). A numerical solution for fractional optimal control problems via Bernoulli polynomials. Journal of Vibration and Control, 22(18), 3889–3903.

11. A.Lotfi, S.A.Yousefi and M. Dehghan, Numerical solution of a class of fractional optimal control problems via the Legendre orthonormal basis combined with the operational matrix and the Gauss quadrature rule, Journal of Computational and Applied Mathematics Volume 250, 2013, 143–160.

12. Lotfi A, Yousefi SA and Dehghan M (2013) Numerical solution of a class of fractional optimal control problems via the Legendre orthonormal basis combined with the operational matrix and the Gauss quadrature rule. Journal of Computational and Applied Mathematics 250: 143–160.

13. Kreyszig E (1978) Introductory Functional Analysis with Applications. New York: John Wiley and Sons.

14. Arfken G (1985) Mathematical Methods for Physicists, 3rd edn. San Diego, CA: Academic Press.

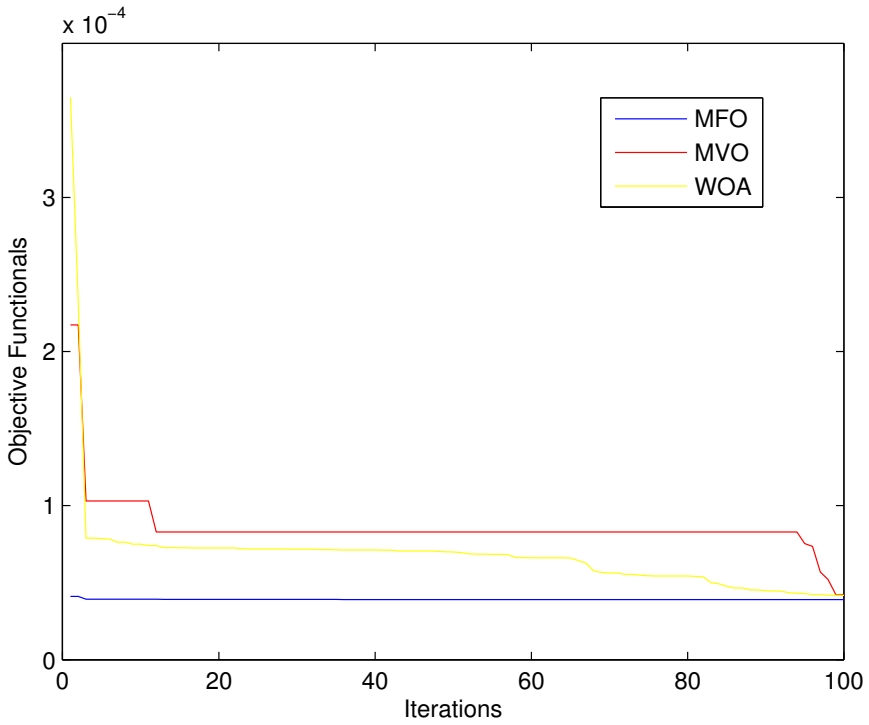

FIGURE 9 Convergence behaviors of the MFO, MVO and WOA algorithms when solving Example 3.

15. Diethelm K and Ford NJ (2004) MultiâĂŞorder fractional differential equations and their numerical solution. Applied Mathematics and Computation 154: 621–640.

16. Costabile F, Dellaccio F and Gualtieri MI (2006) A new approach to Bernoulli polynomials. Rendiconti di Matematica, Serie VII 26: 1–12.

17. Kreyszig E (1978) Introductory functional analysis with applications.Wiley, New York

18. Keshavarz, E., Ordokhani, Y. and Razzaghi, M. (2016). A numerical solution for fractional optimal control problems via Bernoulli polynomials. Journal of Vibration and Control, 22(18), 3889–3903.

19. Kreyszig E (1978) Introductory functional analysis with applications.Wiley, New York

20. G. Arfken, Mathematical Methods for Physicists, third ed., Academic press, San Diego, 1985.

21. S. Mirjalili, S.M. Mirjalili, and A. Hatamlou, Multi-verse optimizer: a nature-inspired algorithm for global optimization, *Neural Computing and Applications*, 2015, doi: 10.1007/s00521-015-1870-7

22. S. Mirjalili, Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm, *Knowledge-Based Systems*, 89(2015): 228–249.

23. S. Mirjalili and A. Lewis, The whale optimization algorithm, *Advances in Engineering Software*, 95 (2016): 51–67.

24. Faris, H., Hassonah, M. A., Ala'M, A. Z., Mirjalili, S., & Aljarah, I. (2018). A multi-verse optimizer approach for feature selection and optimizing SVM parameters based on a robust system architecture. *Neural Computing and Applications*, 30(8), 2355–2369.

25. Fathy, A., & Rezk, H. (2018). Multi-verse optimizer for identifying the optimal parameters of PEMFC model. *Energy*, 143, 634–644.

26. Jangir, P., Parmar, S. A., Trivedi, I. N., & Bhesdadiya, R. H. (2017). A novel hybrid particle swarm optimizer with multi verse optimizer for global numerical optimization and optimal reactive power dispatch problem. *Engineering Science and Technology, an International Journal*, 20(2), 570–586.

27. Yildiz, B. S., & Yildiz, A. R. (2017). Moth-flame optimization algorithm to determine optimal machining parameters in manufacturing processes. *Materials Testing*, 59(5), 425–429.

28. Mei, R. N. S., Sulaiman, M. H., Mustaffa, Z., & Daniyal, H. (2017). Optimal reactive power dispatch solution by loss minimization using moth-flame optimization technique. *Applied Soft Computing*, 59, 210–222.

29. Khalilpourazari, S., & Khalilpourazary, S. (2019). An efficient hybrid algorithm based on Water Cycle and Moth-Flame Optimization algorithms for solving numerical and constrained engineering optimization problems. *Soft Computing*, 23(5), 1699–1722.

30. Aljarah, I., Faris, H., & Mirjalili, S. (2018). Optimizing connection weights in neural networks using the whale optimization algorithm. *Soft Computing*, 22(1), 1–15.

31. Mafarja, M., & Mirjalili, S. (2018). Whale optimization approaches for wrapper feature selection. *Applied Soft Computing*, 62, 441–453.

32. Abdel-Basset, M., Manogaran, G., El-Shahat, D., & Mirjalili, S. (2018). A hybrid whale optimization algorithm based on local search strategy for the permutation flow shop scheduling problem. *Future Generation Computer Systems*, 85, 129–145.