

RESEARCH ARTICLE

Optimal control of fractional integro-differential systems based on a spectral method and grey wolf optimizer

Raheleh Khanduzi^{*a}, Asyieh Ebrahimzadeh^b, and Samaneh Panjeh Ali Beik^c

^aDepartment of Mathematics and Statistics, Gonbad Kavous University, Golestan, Iran

^bSchool of Basic Sciences, Farhangian University, Tehran, Iran

^cYoung Researchers and Elite Club, Karaj Branch, Islamic Azad University, Karaj, Iran
khanduzi@gonbad.ac.ir, ebrahimzadeh263@gmail.com, panjehali@alumni.iust.ac.ir

ARTICLE INFO

Article History:

Received 16 November 2018

Accepted 24 July 2019

Available 14 January 2020

Keywords:

Optimal control

Fractional Volterra integro-differential equation

Collocation method

Grey wolf optimizer

AMS Classification 2010:

49J21; 65N35; 97R40

ABSTRACT

This paper elaborated an effective and robust metaheuristic algorithm with acceptable performance based on solution accuracy. The algorithm applied in solution of the optimal control of fractional Volterra integro-differential (FVID) equation which be substituted by nonlinear programming (NLP). Subsequently the FIVD convert the problem to a NLP by using spectral collocation techniques and thereafter we execute the grey wolf optimizer (GWO) to improve the speed and accuracy and find the solutions of the optimal control and state as well as the optimal value of the cost function. It is mentioned that the utilization of the GWO is simple, due to the fact that the GWO is global search algorithm, the method can be applied to find optimal solution of the NLP. The efficiency of the proposed scheme is shown by the results obtained in comparison with the local methods. Further, some illustrative examples introduced with their approximate solutions and the results of the present approach compared with those achieved using other methods.



1. Introduction

The main purpose of this essay is to introduce an efficient approach for solving following optimal control problem (OCP):

Problem A: Find optimal control u^* and corresponding optimal state x^* that minimize the quadratic performance index

$$J = \int_0^T (x^2(t) + u^2(t) + f(t)x(t) + g(t)u(t)) dt, \quad (1)$$

subject to the fractional Volterra integro-differential (FVID) equation

$$D^\alpha x(t) = a(t)x(t) + b(t)u(t) + \int_0^t (K(t,s))\varphi(x(s))ds, \quad (2)$$

where $a(t)$, $b(t)$, $g(t)$, $f(t)$ are known and real valued functions which are belonged to $L^2[0, T]$ and $\varphi(x(s))$ is a nonlinear function in terms of the unknown function $x(s)$.

In various problems of physics, mechanics and engineering, fractional differential equations have been proved to be a valuable tool in the modeling of many phenomena. There are many applications in viscoelasticity, electrochemistry, control and electromagnetic, [1, 2]. In consequence, the subject of fractional equations is gaining much importance and attention. Meanwhile, the study of OCP governed by fractional integro differential equations is also important as such systems occur in various problems of applied nature. Some approaches for numerical solutions of fractional optimal control problems can be found in [3–6].

*Corresponding Author

We need to be mindful only special cases of OCPs can be solved analytically, so choosing the best numerical schemes in terms of rapidity of convergence and accuracy is significant. The method implemented for discretizing mentioned OCP is spectral method, which is one of the most accurate method which used by several author and in different kind of functional problems for example see the [7,8] and the references in. The idea is to write the solution of OCP as a sum of Bernoulli polynomials, substituting these approximations in the OCP yields a NLP in the coefficients which can be solved using any metaheuristic algorithm presented in the literature for solving an optimization problem.

Among these algorithms, nature-inspired metaheuristic algorithms are appropriate for global searches according to ability in exploring globally and exploiting locally. Mirjalili et al. [9] proposed grey wolf optimizer (GWO) algorithm inspired by the behavior of grey wolves in nature. Indeed, the GWO algorithm simulated the leadership hierarchy and hunting behavior of grey wolves. GWO has shown a good performance when applied to solve nonlinear continuous optimization problems. The GWO algorithm is also compared with particle swarm optimization, gravitational search algorithm, differential evolution, evolutionary programming, and evolution strategy to confirm its results.

So, GWO is theoretically able to solve our NLP. Some points on the advantages of the GWO have been expressed:

- The social hierarchy helped GWO to visit the best solutions generated over the course of iteration.
- The encircling procedure determined a circle-shaped neighborhood around the solutions which can be developed to higher dimensions as a hyper-sphere.
- The random parameters helped candidate solutions to have hyper-spheres with different random radii.
- The hunting approach accepted candidate solutions to detect the probable location of the prey.
- Exploration and exploitation are warranted by the adaptive values of two parameters.
- The adaptive values of parameters helped GWO to efficiently trade off between exploration and exploitation.
- The GWO had only two main parameters to be controlled.

The paper is organized as follows: In section 2, the basic concepts about the Bernoulli polynomials and how to approximate the functions in terms of these polynomials is interpreted. Also, the operational matrices of fractional integration are mentioned. As we have provided some definition of fractional calculus. In section 3, the outline of our spectral scheme for discretizing aforementioned optimal control problem and obtaining the resulted NLP is presented. Section 4 is devoted to explain the grey wolf optimizer algorithm for solving the problem under consideration.

In section 5, numerical results are reported to verify the applicability of the presented method in comparison with the other methods in the literature. Through these examples, the superiority of these three bases functions are also discussed. Finally, section 6 ends this paper with a brief conclusion and some remarks.

2. Preliminaries

In this section, we give some basic concepts we require.

2.1. Fractional Calculus

This section, reviews some basic definitions and notations of fractional integral and derivative which are applied further in this work [10].

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

$$I^\alpha \xi(t) = \frac{1}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1} \xi(s) ds, \quad \alpha > 0, t > 0. \quad (3)$$

in which $\Gamma(\cdot)$ denotes the Gamma function and for $\alpha = 0$, we set $I^0 \xi(t) = \xi(t)$.

Definition 2. *Let $n = \lceil \alpha \rceil$ ($\lceil \cdot \rceil$ denotes ceiling function, $\lceil t \rceil = \min\{z \in Z : z \geq t\}$), the operator D^α , defined by*

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

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

The one type of fractional derivative is Caputo fractional derivative, which is frequently used in applications.

Definition 3. *The Caputo fractional derivative of f , is defined as*

$$D_*^\alpha \xi(t) = \begin{cases} I^{n-\alpha} D^n \xi(t), & n-1 < \alpha < n, n \in \mathbb{N}, \\ \frac{d^n}{dt^n} \xi(t), & \alpha = n. \end{cases} \quad (4)$$

Lemma 1. Let $\alpha, \beta \geq 0$, $c_1, c_2 \in \mathfrak{R}$ and $f(t), g(t) \in L_1[0, T]$. Then

- 1) $I^\alpha I^\beta f(t) = I^\beta I^\alpha f(t)$,
- 2) $I^\alpha I^\beta f(t) = I^{\beta+\alpha} f(t)$,
- 3) $D^\alpha(c_1 f(t) + c_2 g(t)) = c_1 D^\alpha(f(t)) + c_2 D^\alpha(g(t))$

Note that for $n-1 < \alpha < n$, $n \in \mathbb{N}$

$$I^\alpha D_*^\alpha \xi(t) = \xi(t) - \sum_{k=0}^{n-1} \xi^{(k)}(0^+) \frac{t^k}{k!}. \quad (5)$$

hold almost everywhere on $[0, T]$.

2.2. An overview on Bernoulli polynomials

Bernoulli polynomials of order m can be defined with the following formula [11],

$$\beta_m(t) = \sum_{i=0}^m \binom{m}{i} \alpha_i t^{m-i}, \quad (6)$$

where α_i , $i = 0, 1, \dots, m$ are Bernoulli numbers. These numbers are a sequence of signed rational numbers which arise in the series expansion of trigonometric functions [12] and can be defined by the identity

$$\frac{t}{e^t - 1} = \sum_{i=0}^m \alpha_i \frac{t^i}{i!}. \quad (7)$$

The first few Bernoulli numbers are

$$\alpha_0 = 1, \quad \alpha_1 = -\frac{1}{2}, \quad \alpha_2 = -\frac{1}{6}, \quad \alpha_4 = -\frac{1}{30}. \quad (8)$$

with $\alpha_{2i+1} = 0$, $i = 1, 2, 3, \dots$. Bernoulli polynomials form a complete basis over the interval $[0, 1]$ [13]. These polynomials satisfy the following formula [12]

$$\int_0^1 \beta_n(t) \beta_m(t) dt = (-1)^{n-1} \frac{m!n!}{(m+n)!} \alpha_{n+m}, \quad (9)$$

$m, n \geq 1$

The first few Bernoulli polynomials are

$$\beta_0(t) = 1,$$

$$\beta_1(t) = t - \frac{1}{2},$$

$$\beta_2(t) = t^2 - t + \frac{1}{6},$$

$$\beta_3(t) = t^3 - \frac{3}{2}t^2 + \frac{1}{2}t.$$

Presume that $H := L^2[0, 1]$ and

$$Y = \text{span}\{\beta_0, \beta_1, \dots, \beta_m\},$$

where $m \in \mathbb{N} \cup \{0\}$ and β_i 's are the Bernoulli polynomials. Since $Y \subset H$ is a finite dimensional vector space, for every $f \in H$, there exists a unique $y_0 \in Y$ such that

$$\|f - y_0\|_2 \leq \|f - y\|_2 \quad \forall y \in Y,$$

in which $\|f\|_2 = \sqrt{\langle f, f \rangle}$. Here, the function y_0 is called the best approximation to f out of Y . As $y_0 \in Y$, we may conclude that

$$f(t) \approx y_0(t) = \sum_{j=0}^m c_j \beta_j(t) = C^T \Psi(t),$$

where

$$\Psi^T(t) = (\beta_0(t), \beta_1(t), \dots, \beta_m(t)), \quad (10)$$

and

$C^T = (c_0, c_1, \dots, c_m)$ such that C uniquely calculated by

$$C = Q^{-1} \int_0^1 f(t) \Psi(t) dt, \quad (11)$$

where $Q \in \mathbb{R}^{(m+1) \times (m+1)}$ is said the dual matrix of $\Psi(t)$ and given by

$$Q = \int_0^1 \Psi(t) \Psi^T(t) dt.$$

For more details about best approximation see [13].

2.3. Bernoulli operational matrix of the fractional integration

In recent years, the operational matrices have attracted researchers attention and applied to solving problems consisted of continuous operators (such as integral, derivative, delay, etc.). Moreover, the numerical methods via these operational matrices are easily implemented and have the following characteristics:

- ◇ play a significant role as a preconditioner in inverse problems,
- ◇ have higher accuracy due to their sparsity.

The RiemannLiouville fractional integration of the vector $\Psi(t)$ given in Equation (10) can be expressed by [4]

$$I^\gamma \Psi(t) = F^\gamma \Psi(t), \tag{12}$$

in which F^γ is the $(m+1) \times (m+1)$ RiemannLiouville fractional operational matrix of integration. Although F^γ given in [4] we use the different way and notations to show this matrix. For this purpose, Assume that S

$$\begin{aligned} I^\gamma \beta_i(t) &= I^\gamma \left(\sum_{k=0}^i \binom{i}{k} \alpha_k t^{i-k} \right) \\ &= \sum_{k=0}^i \binom{i}{k} \alpha_k I^\gamma t^{i-k} \\ &= \sum_{k=0}^i \frac{\binom{i}{k} \Gamma(i-k+1) \alpha_k}{\Gamma(i-k+1+\gamma)} t^{i-k+\gamma} \end{aligned} \tag{13}$$

Now if $t^{i-k+\gamma}$ approximated in terms of Bernoulli polynomials we can define each elements of $S = [s_{ij}]_{(m+1) \times (m+1)}$ as

$$s_{ij} = \sum_{k=0}^i \sum_{l=0}^j \frac{\Gamma(i-k+1) \binom{i}{k} \binom{j}{l} \alpha_k \alpha_l}{\Gamma(i-k+1+\gamma) \Gamma(-l-k+\alpha+1+i+j)}, \quad i, j = 0, 1, \dots, m. \tag{14}$$

As a results F^γ can be expressed as

$$F^\gamma = SQ^{-1} \tag{15}$$

3. Bernoulli polynomial collocation method

For discretization of the integro-differential dynamic system (2), we express the fractional state rate $D^\gamma x(t)$ and control variable $u(t)$ in terms of Bernoulli polynomial as

$$\begin{aligned} D^\gamma x(t) &\simeq X^T \Psi(t), \\ u(t) &= U^T \Psi(t), \end{aligned} \tag{16}$$

where X^T and U^T are unknown vectors and $\Psi(t)$ given in (10). Using Lemma 2.1. Equation (12), $x(t)$ can be represented by

$$x(t) = I^\gamma D^\gamma x(t) + x(0) \simeq (X^T F^\gamma + E^T) \Psi(t). \tag{17}$$

F^γ is the fractional operational matrix of integration and $E^T = [x_0, 0, \dots, 0]_{(1 \times (m+1))}$. Now we replace (16) and (17) in dynamic system (2)

$$\begin{aligned} X^T \Psi(t) - a(t)(X^T F^\gamma + E^T) \Psi(t) - b(t)U^T \Psi(t) \\ - \int_0^t k(t, s) \varphi((X^T F^\gamma + E^T) \Psi(s)) ds = 0, \end{aligned} \tag{18}$$

In order to specify the unknown coefficients in (18), we collocate this equation at $m+1$ collocation points. So (18) can be rewrite as

$$\begin{aligned} X^T \Psi(t_i) - a(t_i)(X^T F^\gamma + E^T) \Psi(t_i) - b(t_i)U^T \Psi(t_i) \\ - \int_0^{t_i} k(t_i, s) \varphi((X^T F^\gamma + E^T) \Psi(s)) ds = 0. \end{aligned} \tag{19}$$

In above equation, $t_i, i = 0, \dots, m$ are the Chebyshev-Gauss-Lobatto nodes in $[0, 1]$ which we chose them as suitable collocation points. In order to utilize the Gauss-Legendre (GL) quadrature formula, by means of transformation $s = \frac{t_i}{2}(\tau + 1)$, (19) convert to

$$\begin{aligned} X^T \Psi(t_i) - a(t_i)(X^T F^\gamma + E^T) \Psi(t_i) \\ - b(t_i)U^T \Psi(t_i) - \frac{t_i}{2} \sum_{j=0}^N \omega_j k(t_i, \frac{t_i}{2}(\tau_j + 1)) \\ \times \varphi((X^T F^\gamma + E^T) \Psi(\frac{t_i}{2}(\tau_j + 1))) = 0, \end{aligned} \tag{20}$$

where τ_j s are GL nodes, zeros of Legendre polynomials in the interval $[-1, 1]$ and ω_j s are the corresponding weights. Although explicit formulas for quadrature nodes are not known, the weights can be expressed in closed form by the following relation consequently, the controlled FVID (2) is reduced to $m+1$ nonlinear algebraic.

For discretization of the performance index stated in (1), we approximate $f(t)$ and $g(t)$ by Bernoulli polynomials respectively as

$$f(t) = F^T \Psi(t), \quad g(t) = G^T \Psi(t). \tag{21}$$

Substituting (21) in (1) conclude that

$$\begin{aligned} J = \int_0^1 (X^T \Psi(t) \Psi^T(t) X + U^T \Psi(t) \Psi^T(t) U \\ + F^T \Psi(t) \Psi^T(t) X + G^T \Psi(t) \Psi^T(t) U) dt \end{aligned} \tag{22}$$

Integrating (21) on $[0, 1]$ results

$$J = X^T Q X + U^T Q U + F^T Q X + G^T Q U, \quad (23)$$

in which Q given in (11). So, The OCP given in (1) and (23) is converted to a NLP with objective functional (22) and constraints (20).

The resulted NLP problem is also large scale. So, it is of great importance to use an efficacious and compatible metaheuristic algorithm which generates the solutions with high computational decisions. This research utilizes a new metaheuristic algorithm, called grey wolf optimizer (GWO) to solve the problem under consideration. The next section describe the GWO and its elements and mechanisms to solve NLP governed by OCP.

4. Grey wolf optimizer

The proposed mathematical programming problem is the nonlinear and large scale. So, we need to solve the problem by studying both local and metaheuristic approaches. Among the local algorithms, the trust region method plays a vital role in solving large-scale nonlinear optimization problems because of its efficiency [14, 15]. However, it finds local solutions in a long amount of time as the $T \gg 1$ or the problem dimension increases. To prevent this type of imperfection, a grey wolf optimizer (GWO) algorithm is proposed in this research. The GWO algorithm is a nature-inspired metaheuristic algorithm which mimitates the leadership hierarchy and hunting structure of grey wolves. Therefore, high-performing metaheuristic approach with high computational high-precision numerical solutions and short execution time is implemented to solve the problem. The GWO obtains near-optimal solutions or the global minimum of objective functional in more efficient way. About the proposed algorithm, it is necessary to note that the GWO is really suitable and appropriate for the nonlinear optimization problems with the number of more variables and constraints, specially when solving large-sized instances of the problems [16–19]. For the problem, the values of objective functionals show that GWO's performance is better than local method in terms of the approximate solution of functions $x(t)$ and $u(t)$. So, on the base of above-mentioned points, one can come to the conclusion that the GWO is a favorable candidate for solving the problem if $T \gg 1$.

The GWO algorithm has derivation-free procedures. In contrast to gradient-based optimization

algorithms, this metaheuristic algorithm minimizes the problems stochastically. The optimization mechanism begins with random solution, and there is no need to compute the derivative of search regions or gradient information of the objective functionals to obtain the global minimum of the problem. This makes the GWO algorithm highly applicable for the NLP problems with unknown derivative information. On the other hand, the simplicity of the GWO is particularly advantageous in the presence of non-smooth objective functionals, for which exact algorithms may fail to reach their global solutions. Viability of the GWO is analyzed using some non-smooth mathematical functions and engineering design problems [20–22]. So, the GWO algorithm is a favorable choice and a competitive algorithm when considering non-smooth, and non-linear functions.

In this section, the essential nature of the GWO algorithm is explained. GWO algorithm is a new nature-inspired metaheuristic algorithm which was first introduced by Mirjalili et al. [9].

GWO is a technique inspired from the nature and grey wolves. The GWO algorithm simulated the leadership hierarchy and hunting behavior of grey wolves.

In leadership hierarchy, alpha, beta, delta, and omega were applied as four grey wolves. Also, hunting, searching for prey, encircling prey, and attacking prey were as the three main components of GWO. These new steps are discussed in the following section.

4.1. Social hierarchy

To formulate the social hierarchy of wolves, the fittest solution is considered as the alpha (α). Accordingly, the second and third best solutions are called beta (β) and delta (δ), respectively. The remaining candidate solutions are then represented as omega (ω). Also, the hunting mechanism is constructed by α , β , and δ . The ω wolves followed these three wolves.

4.2. Encircling prey

As seen in nature, grey wolves surround prey during the hunt. This surrounding behavior is given by:

$$\vec{D} = |\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)| \quad (24)$$

$$\vec{X}(t+1) = \vec{X}(t) - \vec{A} \cdot \vec{D} \quad (25)$$

where t shows the current iteration, \vec{A} and \vec{C} are coefficient vectors, \vec{X}_p is the position vector of the

prey, and \vec{X} shows the position vector of a grey wolf. The vectors \vec{A} and \vec{C} are computed followed by:

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a} \quad (26)$$

$$\vec{C} = 2 \cdot \vec{r}_2 \quad (27)$$

where r_1 and r_2 are random vectors in $[0, 1]$. Over the course of iterations, components of \vec{a} are linearly reduced from 2 to 0.

4.3. Hunting

Grey wolves have the capability to identify the position of prey and envelop them. The hunt is normally conducted by the alpha. The beta and delta also partake in hunting sometimes. So, the first three best solutions is saved and the other search agents (including the omegas) is performed to update their positions based on the position of the best search agents. This process is stated with the following equations:

$$\vec{D}_\alpha = |\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}|, \vec{D}_\beta = |\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}|, \vec{D}_\delta = |\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}|, \quad (28)$$

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot (\vec{D}_\alpha), \vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot (\vec{D}_\beta), \vec{X}_3 = \vec{X}_\delta - \vec{A}_3 \cdot (\vec{D}_\delta), \quad (29)$$

$$\vec{X}_{t+1} = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (30)$$

4.4. Attacking prey

At the end of the hunt, grey wolves rush at the prey when it stops moving. To model nearing the prey, the value of \vec{a} is reduced from 2 to 0. Then, the variation range of \vec{A} is also reduced by \vec{a} . Especially, \vec{A} is a random value in the interval $[-2a, 2a]$.

4.5. Search for prey

Grey wolves chiefly explore based on the position of the alpha, beta, and delta. They get away from each other to search for prey and converge to rush prey. To formulate divergence, \vec{A} with random values greater than 1 or less than -1 is applied to enforce the search agent for diverging from the prey. So, the GWO algorithm employing global search strategy and this confirms exploration.

As it is seen in Eq. (27), the \vec{C} vector is also another factor of exploration. This factor obtains random weights for prey to stochastically accentuate ($C > 1$) or unaccentuate ($C < 1$) the efficacy

of prey in determining the distance in Eq. (24). The C vector can be also considered as the efficacy of barriers to nearing prey in nature. Usually, the barriers in nature exist in the hunting paths of wolves and impede them from swiftly and comfortably nearing prey.

Briefly, the search procedure starts with generating a random population of grey wolves. Over the course of iterations, alpha, beta, and delta wolves suggest the possible location of the prey. The distance from the prey is updated by each candidate solution. The parameter a is reduced from 2 to 0 to accentuate exploration and exploitation, respectively.

Candidate solutions favor divergence of the prey when $|\vec{A}| > 1$ and move towards (converge) the prey when $|\vec{A}| < 1$. Finally, the GWO algorithm stops when an end criterion is satisfied.

The pseudo code of the GWO algorithm is presented in Algorithm 1.

Algorithm 1. GWO algorithm

- 1: Initialize the grey wolf population $X_i = (x_i(t), u_i(t)) (i = 1, \dots, n)$
- 2: Initialize a , A , and C
- 3: Calculate the fitness of each search agent
- 4: X_α = the best search agent
- 5: X_β = the second best search agent
- 6: X_δ = the third best search agent
- 7: **while** ($k < \text{Max number of iterations}$) **do**
- 8: **for** each search agent **do**
- 9: Update the position of the current search agent by equation (30)
- 10: **end for**
- 11: Update a , A , and C
- 12: Calculate the fitness of all search agents
- 13: Update X_α , X_β , and X_δ
- 14: $k = k + 1$
- 15: **end while**
- 16: return X_α

5. Numerical experiments

In these examples, firstly the OCP is converted to a NLP with proposed method in section 3. The resulted NLP is solved by using *fmincon* function in MATLAB and GWO algorithm to find local and global minimum of constrained nonlinear function, respectively.

For ease of references in tables, we use the proposed method and local method to demonstrate the results obtained from solving NLP with *fmincon* function and GWO algorithm, respectively.

In order to demonstrate and justify the performance and the accuracy of our scheme on OCPs governed by fractional integro-differential equation, we consider the following examples.

Example 1. Consider the following OCP

$$J = \int_0^1 ((x(t) - e^t)^2 + (u(t) - e^{3t})^2) dt \quad (31)$$

subject to the nonlinear fractional integro-differential equation

$$D^\alpha x(t) - \frac{3}{2}x(t) + \frac{1}{2}u(t) - \int_0^t (e^{t-s}x^3(s)) ds = 0, \quad x(0) = 1. \quad (32)$$

The problem is to find the optimal control $u^*(t)$, which minimizes the quadratic performance index (31). For this problem, the exact solution in the case of $\alpha = 1$ is given by [23]

$$x(t) = e^t, \quad u(t) = e^{3t}.$$

In Table 1, one can compare the optimal value of objective functional by utilizing GOW algorithm as well as local method in $M = 7$ and different values of α . The numerical results for $x(t)$ and $u(t)$ in $M = 7$ and $\alpha = 0.5, 0.7, 0.9$ and 1 are plotted in Figures 1-2. In these figures, we see that our approximate solutions converge to exact solution. The results obtained with GOW method demonstrate validity and effectiveness of proposed method.

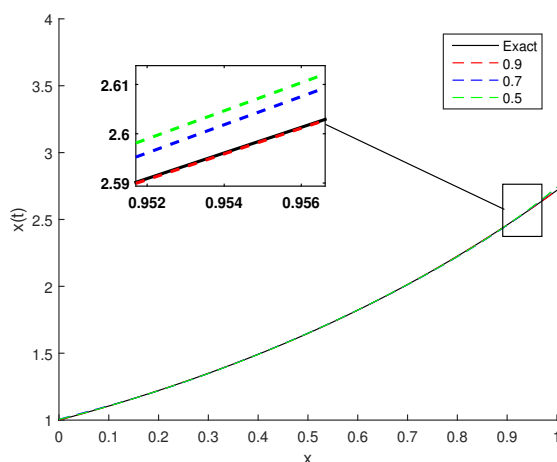


Figure 1. State $x(t)$ as a function of t for the Example 1 for $m = 7$ and different values of α (green: $\alpha = 0.5$, blue: $\alpha = 0.7$, red: $\alpha = 0.9$).

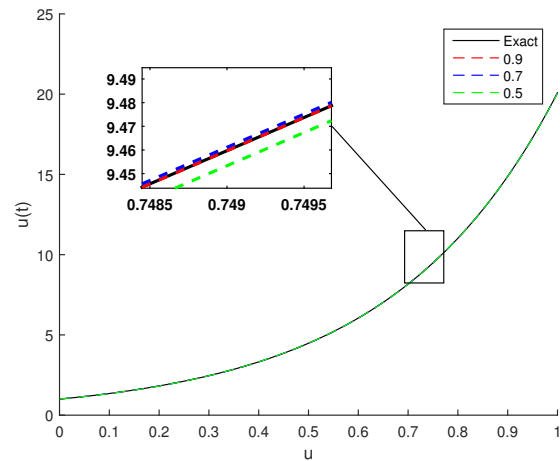


Figure 2. Control $u(t)$ as a function of t for the Example 1 for $m = 7$ and different values of α (green: $\alpha = 0.5$, blue: $\alpha = 0.7$, red: $\alpha = 0.9$).

Example 2. Consider the following nonlinear problem [23]

$$\min J = \int_0^1 ((x(t) - e^{t^2})^2 + (u(t) - (1 + 2t))^2) dt,$$

$$D^\alpha x(t) + x(t) - u(t) - \int_0^t (t(1 + 2t)e^{s(t-s)}x(s)) ds = 0$$

The optimal control u^* and corresponding optimal state x^* for $\alpha = 1$ are respectively $1 + 2t$ and e^{t^2} .

Figures 3-4 show the approximate solution of functions $x(t)$ and $u(t)$ using GWO algorithm and local method for $M = 7$ and $\alpha = 0.5, 0.7, 0.9$. The exact solution for $\alpha = 1$ is also represented. The value of objective function with GWO and local methods for $M = 7$ and different values of α are given in Table 2. It is obvious that we can achieve a better approximation with GWO algorithm against local method.

Table 1. The value of J^* for Example 1 ($m = 7$ and different α).

Local Method			Proposed Method			
$\alpha = 0.7$	$\alpha = 0.9$	$\alpha = 1$	$\alpha = 0.7$	$\alpha = 0.9$	$\alpha = 1$	
J^*	0.389877	0.0946404	4.17536×10^{-11}	2.45×10^{-5}	8.64×10^{-8}	7.74×10^{-12}

Table 2. The value of J^* for Example 2 ($m = 7$ and different α).

Local Method			Proposed Method			
$\alpha = 0.7$	$\alpha = 0.9$	$\alpha = 1$	$\alpha = 0.7$	$\alpha = 0.9$	$\alpha = 1$	
J^*	0.0515124	0.00494284	1.29×10^{-11}	2.50×10^{-6}	1.22×10^{-8}	4.70×10^{-12}

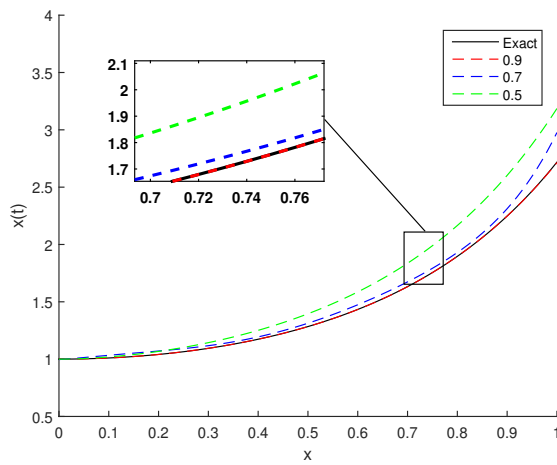


Figure 3. State $x(t)$ as a function of t for the Example 2 for $m = 7$ and different values of α (green: $\alpha = 0.5$, blue: $\alpha = 0.7$, red: $\alpha = 0.9$).

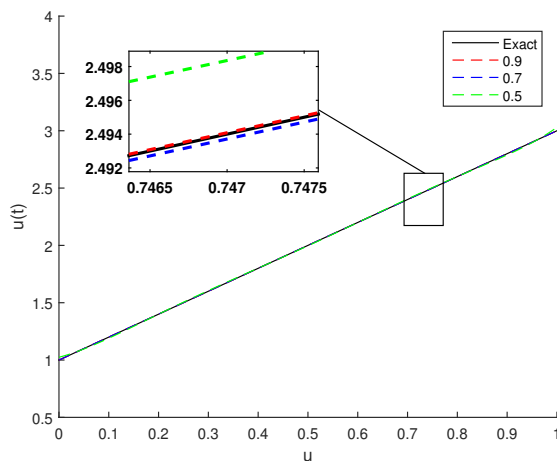


Figure 4. Control $u(t)$ as a function of t for the Example 2 for $m = 7$ and different values of α (green: $\alpha = 0.5$, blue: $\alpha = 0.7$, red: $\alpha = 0.9$).

Example 3. Consider the minimization of fractional [23]

$$J = \int_0^1 \left((x(t) - t)^2 + (u(t) - te^{t^2})^2 \right) dt,$$

subject to dynamic state

$$D^\alpha x(t) - x(t) - u(t) + 2 \int_0^t \left(tse^{-x^2(s)} \right) ds = 0.$$

The optimal control $u^*(t)$ and corresponding optimal state $x(t)$ for $\alpha = 1$ are as follows:

$$\begin{aligned} x^*(t) &= t, \\ u^*(t) &= 1 - te^{-t^2} \end{aligned}$$

We solve this OCP using GWO and local methods for for $M = 7$ and various α . Figures 5-6 show that as $\alpha \rightarrow 1$, the approximate solutions with GWO algorithm tend to the exact solution in the case of $\alpha = 1$. The value of objective function with GWO and local methods for $M = 7$ and different values of α , is shown in Table 3. From Table 3, we can see that the value of objective function based on GWO is all less than the least value of objective function obtained by local method.

Table 3. The value of J^* for Example 3 ($m = 7$ and different α).

Local Method			Proposed Method			
$\alpha = 0.7$	$\alpha = 0.9$	$\alpha = 1$	$\alpha = 0.7$	$\alpha = 0.9$	$\alpha = 1$	
J^*	0.0366084	0.00374084	8.84×10^{-14}	4.71×10^{-7}	9.37×10^{-8}	2.12×10^{-14}

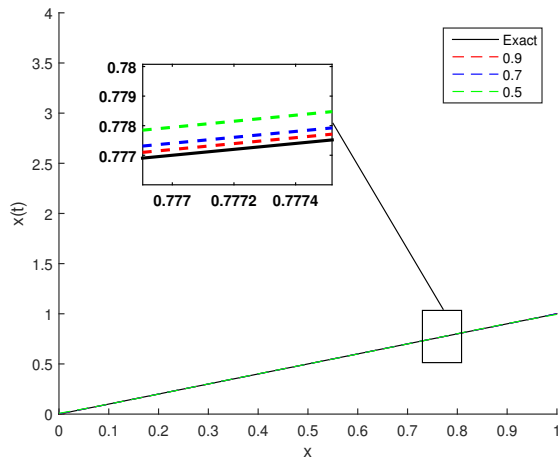


Figure 5. State $x(t)$ as a function of t for the Example 3 for $m = 7$ and different values of α (green: $\alpha = 0.5$, blue: $\alpha = 0.7$, red: $\alpha = 0.9$).

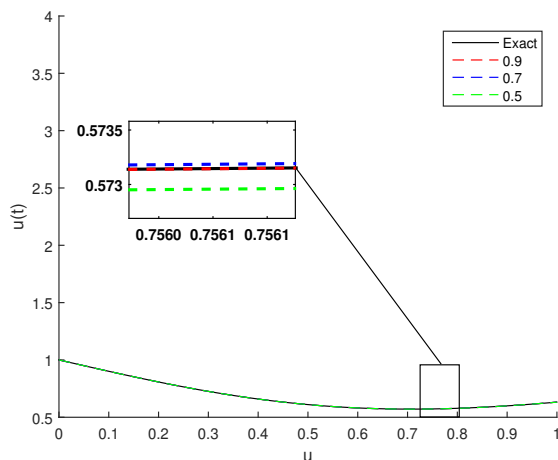


Figure 6. Control $u(t)$ as a function of t for the Example 3 for $m = 7$ and different values of α (green: $\alpha = 0.5$, blue: $\alpha = 0.7$, red: $\alpha = 0.9$).

6. Conclusions

By utilizing spectral method, OCP governed by fractional Volterra-integro differential equation is converted to a NLP.

In this research, a powerful and efficacious meta-heuristic algorithm called Grey Wolf Optimizer (GWO) is utilized to obtain the solutions of the optimal control and state as well as the optimal value of the objective function.

The GWO algorithm imitated the leadership hierarchy with four types of grey wolves and hunting procedure with searching for prey, encircling prey, and attacking prey. These strategies confirmed the preferable exploitation, exploration capability and efficient escape from local optimum of the GWO.

Numerical experiments verify the validity and the applicability of the proposed method. Comparisons with the exact solution and other methods show that this technique is a powerful and efficient tool for solving the fractional OCP.

Acknowledgments


The first author would like to thank Gonbad Kavous University for supporting this research work. The second author would like to appreciate the research council of Farhangian University for supporting this research. The work 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] Diethelm, K., Ford, N.J. and Freed, A.D. (2004). Detailed error analysis for a fractional Adams method. *Numerical Algorithms*, 36, 31–52.
- [2] Huang, L., Li, X.F., Zhao, Y.L. and Duan, X.Y. (2011). Approximate solution of fractional integro-differential equations by Taylor expansion method. *Computers & Mathematics with Applications*, 62(3), 1127–1134.
- [3] Tohidi, E. and Nik, H.S. (2015). A bessel collocation method for solving fractional optimal control problems. *Applied Mathematical Modelling*, 39(2), 455–465.
- [4] Keshavarz, E., Ordokhani, Y. and Razzaghi, M. (2015). A numerical solution for fractional optimal control problems via Bernoulli polynomials. *Journal of Vibration and Control*, 22(18), 3889–3903.
- [5] Zaky, M.A. and Machado, J.A.T. (2017). On the formulation and numerical simulation of distributed-order fractional optimal control problems. *Communications in Nonlinear Science and Numerical Simulation*, 52, 177–189.

- [6] Salati, A.B., Shamsi, M., Torres, D.F.M. (2019). Direct transcription methods based on fractional integral approximation formulas for solving nonlinear fractional optimal control problems. *Communications in Nonlinear Science and Numerical Simulation*, 67, 334–350.
- [7] Youssri, Y.H., and Abd-Elhameed, W.M. (2018). Spectral tau algorithm for solving a class of fractional optimal control problems via Jacobi polynomials. *An International Journal of Optimization and Control: Theories & Applications (IJOCTA)*, 8(2), 152–160.
- [8] Youssri, Y.H. and Hafez, R.M. (2019). Chebyshev collocation treatment of Volterra-Fredholm integral equation with error analysis. *Arabian Journal of Mathematics*, 1–10.
- [9] Mirjalili, S.A., Mirjalili, S.M. and Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software*, 69, 46–61.
- [10] Baleanu, D., Diethelm, K., Scalas, E. and Trujillo, J.J. (2012). **Fractional calculus models and numerical methods**. Nonlinearity and Chaos, Series on Complexity, World Scientific.
- [11] Costabile, F., Dellaccio, F. and Gualtieri, M.I. (2006). A new approach to Bernoulli polynomials. *Rendiconti di Matematica, Serie VII*, 26, 1–12.
- [12] Arfken G. (1985). *Mathematical methods for physicists*. 3rd edn. San Diego, CA: Academic Press.
- [13] Kreyszig, E. (1978). *Introductory functional analysis with applications*. New York: John Wiley and Sons.
- [14] Yuan, Y.X. (2000). A review of trust region algorithms for optimization. *In Iciam*, 99(1), 271–282.
- [15] Sadjadi, S.J. and Ponnambalam, K. (1999). Advances in trust region algorithms for constrained optimization. *Applied Numerical Mathematics*, 29(3), 423–443.
- [16] Faris, H., Aljarah, I., Al-Betar, M. A. and Mirjalili, S. (2018). Grey wolf optimizer: a review of recent variants and applications. *Neural computing and applications*, 30(2), 413–435.
- [17] Wang, J.S. and Li, S.X. (2019). An improved grey wolf optimizer based on differential evolution and elimination mechanism. *Scientific reports*, 9(1), 71–81.
- [18] Liu, H., Hua, G., Yin, H. and Xu, Y. (2018). An intelligent grey wolf optimizer algorithm for distributed compressed sensing. *Computational intelligence and neuroscience*, 2018.
- [19] Gupta, S. and Deep, K. (2018). An opposition-based chaotic grey wolf optimizer for global optimisation tasks. *Journal of Experimental & Theoretical Artificial Intelligence*, 1–29.
- [20] Abdo, M., Kamel, S., Ebeed, M., Yu, J. and Jurado, F. (2018). Solving non-smooth optimal power flow problems using a developed grey wolf optimizer. *Energies*, 11(7), 1692.
- [21] Pradhan, M., Roy, P. K. and Pal, T. (2016). Grey wolf optimization applied to economic load dispatch problems. *International Journal of Electrical Power & Energy Systems*, 83, 325–334.
- [22] Tung, N.S. and Chakravorty, S. (2015). Grey Wolf optimization for active power dispatch planning problem considering generator constraints and valve point effect. *International Journal of Hybrid Information Technology*, 8(12), 117–134.
- [23] Maleknejad, K., Ebrahimzadeh, A. (2014). Optimal control of volterra integro-differential systems based on legendre wavelets and collocation method. *Journal of Mathematical, Computational, Physical and Quantum Engineering*, 8(7), 1040–1044.


Rahelah Khanduzi is an Assistant Professor in the Department of Mathematics at Gonbad Kavous University, Gonbad Kavous, Golestan, Iran. She received her Ph.D. in Operations Research from the Faculty of Mathematics, Shiraz University of Technology, Shiraz, Iran, M.Sc. in Operations Research from the Faculty of Mathematics, K.N. Toosi University of Technology, Tehran, Iran and B.Sc. from the Faculty of Mathematics, Shahrood University of Technology, Shahrood, Iran. Her research interests include Linear and Nonlinear Optimization, Mathematical Modeling, Convex Optimization, Combinatorial Optimization, Facility Location Problem, Service Industries or Systems, System Protection, Stackelberg Game, Optimal Control, Metaheuristic Algorithms, Exact Algorithms, Data Mining, and Neural Networks.

 <http://orcid.org/0000-0002-0979-4041>

Asyieh Ebrahimzadeh is an Assistant Professor in the School of Basic Sciences, Farhangian University, Tehran, Iran. She received her Ph.D. in Numerical Analysis from the Department of Applied Mathematics at Iran University of Science and Technology, Tehran, Iran, M.Sc. in Applied Mathematics from the Faculty of Mathematics, K.N. Toosi University of Technology, Tehran, Iran and B.Sc. in Mathematics at the ValiAsr University of Rafsanjan, Kerman, Iran. Her scientific interests include Optimal control, Wavelets, Integral Equations, and Orthogonal functions.

 <http://orcid.org/0000-0002-4684-7640>

Samaneh Panjeh Ali Beik is a Ph.D. in Numerical Analysis from the Department of Applied Mathematics at Iran University of Science and Technology, Tehran, Iran. She received her M.Sc. in Applied Mathematics from the Faculty of Mathematics, Alzahra University, Tehran, Iran. Her scientific interests include Optimal control, Wavelets, Fractional Equations, and Orthogonal functions.

 <http://orcid.org/0000-0002-6559-3279>



This work is licensed under a Creative Commons Attribution 4.0 International License. The authors retain ownership of the copyright for their article, but they allow anyone to download, reuse, reprint, modify, distribute, and/or copy articles in IJOCTA, so long as the original authors and source are credited. To see the complete license contents, please visit <http://creativecommons.org/licenses/by/4.0/>.