Comparative Analysis of the Convergence of the Population-Based Algorithm and the Gradient Algorithm for Optimizing the Neural Network Solution of the Optimal Control Problems

Irina Bolodurina
Orenburg State University
Federal Research Centre of Biological Systems and *Agrotechnologies RAS*
Orenburg, Russia
prmat@mail.osu.ru

Lubov Zabrodina
Department of Applied Mathematics
Orenburg State University
Orenburg, Russia
zabrodina97@inbox.ru

***Abstract*—In this paper, we consider the functional representation of the solution of the optimal control problem without restrictions using the neural network approach, which allows us to find a functional representation of the solution. Based on the necessary first-order optimality conditions, the original problem is reduced to a nonlinear optimization problem where the weights and displacements associated with all neurons are unknown. The minimization of the error function of the neural network solution is carried out by the gradient descent method, as well as by the population gravity search algorithm. Several examples demonstrating the effectiveness of the considered methods are considered. A comparative analysis of the convergence of the algorithms used is carried out. The study showed that the gravitational search algorithm, which requires the least number of iterations to achieve accuracy, is more efficient. Such an effect may be due to the gully of the minimized Lagrange function, as well as with other factors that require additional research.**

Keywords—nonlinear optimization problem, neural network, gradient descent method, gravity search algorithm, population algorithm

I. INTRODUCTION

Optimal control of nonlinear systems is one of the most interesting questions in control theory. Even when it is possible to find an analytical expression for an optimal control function, the form of this function is quite complex. Most of the literature on numerical methods for solving General optimal control problems is focused on algorithms for solving discrete problems. The main idea of these methods is to apply nonlinear programming methods to the finite-dimensional optimization problem [1].

In recent years, neural networks have been used to obtain numerical solutions to the optimal control problem. There are already adaptive neural network architectures for optimal control problems with control and state constraints[2,3].

For this reason, this paper discusses the possibility of using neural networks to solve various types of optimal control problems. The ability of neural networks to approximate nonlinear functions is Central to their application in optimization. Therefore, it can be effectively used to represent nonlinear control [4]. However, the question of the convergence of the developed algorithm is still being investigated, which is associated with the possibility of choosing an optimization algorithm for updating the weight coefficients. In this regard, this study analyzes the speed of convergence of the neural network method using population and gradient optimization algorithms for finding weight coefficients.

There are many examples of optimization problems, the solution of which is based on the structure of neural networks [5,6]. In particular, the numerical solution of optimal control problems has a well-developed theoretical basis [7,8].

In the article [9] the authors Suykeyns J. K. and Bersini H consider the construction of neural network solution of optimal control problems, where the solution of a nonlinear system is using dynamic programming and Q-learning methods.

Using trial solutions based on neural network and collocation points, the numerical problem of solving the optimal control problem is transformed into a nonlinear optimization problem. This was presented in more detail by the authors A. Nazemi, R. Karami in their work [10].

It should be noted that the structure of this work differs from [10] in two important aspects. In this study, we consider the problem of optimal control without restrictions. Also, the present study proposes a study of the convergence of the dynamic optimization scheme by modifying the stage of updating the weight coefficients of the population gravity search algorithm and gradient descent algorithm.

II. FORMULATION OF THE OPTIMAL CONTROL PROBLEM

We consider the Boltz problem of finding a control $u(t)$ that minimizes the functional:

$$J(x,u) = \Phi(t,x)\Big|_{t_f} + \int_{t_0}^{t_f} F(t,x,u)dt \to \min \qquad (1)$$

by

$$\dot{x}(t) = f(t,x,u), \qquad (2)$$

$$x(t_0) = x_0 \qquad (3)$$

$$u(t) \in R^m, x(t) \in R^n, \tag{4}$$

where t_0, t_f, x_0 are fixed.

The trajectory $x(t)$ determined by the dynamics (2) for the functional (1) according to the Lagrange multiplier method has conjugate factors $\lambda \in R^n (\lambda \geq 0)$ not equal to zero at the same time and such that the Lagrange function has the form:

$$\begin{aligned} L(t,x,u,\lambda) = \Phi(t,x)\big|_{t_f} + \int_{t_0}^{t_f} (F(t,x,u) \\ + \lambda(f(t,x,u) - \dot{x}))dt \end{aligned} \tag{5}$$

Let's use the necessary conditions of the first order extremum:

$$\dot{x}_i = f_i, \quad i = 1,..,n \tag{6}$$

$$\dot{\lambda}_i = -\frac{\partial F}{\partial x_i} - \sum_{i=1}^{n} \frac{\partial f_i}{\partial x_i} \lambda_i, \quad i = 1,..,n \tag{7}$$

$$0 = \frac{\partial F}{\partial u_s} + \sum_{i=1}^{n} \frac{\partial f_i}{\partial u_s} \lambda_i, \; s = 1,..,m \tag{8}$$

$$x_i(t_0) = x_{0_i}, \quad i = 1,..,n \tag{9}$$

$$\lambda_i(T) = -\frac{\partial \Phi}{\partial x_i}\bigg|_T, \quad i = 1,..,n \tag{10}$$

The nonlinear optimization problem (6)-(10), which is equivalent to the original optimal control problem (1)-(4), provides the possibility of applying the neural network approach.

III. Neural Network Approach For Solving Nonlinear Optimization Problems

Consider in General a two-layer perceptron with n inputs, one hidden layer with sigmoid activation functions and a linear output block.

For a given vector of input signals $t = (t_1, ..., t_k)$ the output of the neural network has the form:

$$N = \sum_{i=1}^{m} v_i \sigma(z_i), \tag{11}$$

and z_i is the total input characteristic of the neural network, having the form:

$$z_i = w_i t + b_i \tag{12}$$

where w_i is the weight parameter of the neuron of the input layer i; v_i is the weight parameter of the neuron of the hidden layer j; b_i means the signal offset of the hidden layer i; $\sigma(z_i)$ is an activation function.

The activation function σ, as a rule, is a one-dimensional non-linear monotonic function. In this study, the sigmoidal activation function of the form will be used:

$$\sigma(x) = \frac{1}{1+e^x}. \tag{13}$$

It is well known that any sufficiently smooth function can be arbitrarily close to a compact set, using a two-layer neural network with appropriate weights. This means that any continuous function can be approximated by a linear combination of sigmoidal functions with any accuracy.

We use this property of neural networks to approximate trajectory functions, Lagrange multipliers and control functions for the optimal control problem (1) - (4). To do this, we consider an approximation scheme for solving the nonlinear optimization problem (6) - (10).

We define neural networks for each function according to the study [10]: the neural network of the trajectory n_x, Lagrange multipliers n_λ and the control n_u in the form:

$$\begin{cases} n_x = \sum_{i=1}^{I} v_x^i \sigma(z_x^i), \; z_x^i = w_x^i t + b_x^i \\ n_\lambda = \sum_{i=1}^{I} v_\lambda^i \sigma(z_\lambda^i), \; z_\lambda^i = w_\lambda^i t + b_\lambda^i \\ n_u = \sum_{i=1}^{I} v_u^i \sigma(z_u^i), \; z_u^i = w_u^i t + b_u^i \end{cases} \tag{14}$$

for $i = 1,..,I$ where I is the number of neurons, which may be different for each neural network.

However, the functions of the form (14) do not take into account the initial condition for the trajectory (9) and the transversality condition for the adjoint factors (10), therefore, we correct the type of solution of the problem (1) - (4), where the boundary conditions are taken into account, in the function of the form:

$$\begin{cases} x_T = x_0 + (t - t_0) n_x, \\ \lambda_T = -\dfrac{\partial \Phi}{\partial x_{i_T}}\bigg|_{t_f} + (t - t_f) n_\lambda, \\ u_T = n_u. \end{cases} \tag{15}$$

The trial solutions (15) are a universal approximation and must satisfy conditions (6) - (10). Due to the fact that conditions (9) - (10) are met when constructing test solutions, we have the following approximation scheme:

$$\dot{x}_{i_T} - f_i = 0, \quad i = 1,..,n \tag{16}$$

$$\dot{\lambda}_{i_T} + \frac{\partial F_T}{\partial x_{i_T}} + \sum_{i=1}^{n} \frac{\partial f_{i_T}}{\partial x_{i_T}} \lambda_{i_T} = 0, \quad i = 1,..,n \tag{17}$$

$$\frac{\partial F_T}{\partial u_{s_T}} + \sum_{i=1}^{n} \frac{\partial f_{i_T}}{\partial u_{s_T}} \lambda_{i_T}, \; s = 1,..,m \tag{18}$$

We divide the interval $[t_0, t_f]$ into points $t_k, k = 1,..,r$ and reduce the solution of the system (16)-(18), using the least squares method, to the problem of minimizing the function $E(y)$:

$$\begin{aligned} \min_y E(y) = \frac{1}{2} \sum_{k=1}^{r} \{E_1(t_k, y) + \\ + E_2(t_k, y) + E_3(t_k, y)\}, \end{aligned} \tag{19}$$

where $y = (w_x, w_\lambda, w_u, b_x, b_\lambda, b_u, v_x, v_\lambda, v_u)^T$ and

$$\begin{cases} E_1(t_k, y) = [\dot{x}_{i_T} - f_i]^2, \\ E_2(t_k, y) = [\dot{\lambda}_{i_T} + \dfrac{\partial F_T}{\partial x_{i_T}} + \sum_{i=1}^{n} \dfrac{\partial f_{i_T}}{\partial x_{i_T}} \lambda_{i_T}]^2, \\ E_3(t_k, y) = [\dfrac{\partial F_T}{\partial u_{s_T}} + \sum_{i=1}^{n} \dfrac{\partial f_{i_T}}{\partial u_{s_T}} \lambda_{i_T}]^2, \end{cases} \quad (20)$$

Here the solution is the vector y, consisting of the weight coefficients of the trial functions having the structure (15).

A. General scheme of neural network algorithm for solving optimization problems:

Preparatory stage:

Put $i = 0$, generate initial weight coefficients:

$$y_i = (w_x, w_\lambda, w_u, b_x, b_\lambda, b_u, v_x, v_\lambda, v_u)^T \quad (21)$$

Calculate the values of the optimized function $E(y_i)$.

1. To update the weight coefficients
$$y_{i+1} = (\tilde{w}_x, \tilde{w}_\lambda, \tilde{w}_u, \tilde{b}_x, \tilde{b}_\lambda, \tilde{b}_u, \tilde{v}_x, \tilde{v}_\lambda, \tilde{v}_u)^T \quad (22)$$
Calculate the values of the optimized function $E(y_{i+1})$.
2. If the stop criterion is reached (for example, $\| E(y_{i+1}) \| < \varepsilon$), then y_{i+1} is the solution of the problem, the end of the algorithm. Otherwise $i = i + 1$ and to step 1.

The convergence of the considered neural network algorithm depends on the optimization method used to update the weight coefficients. In this regard, the convergence of the neural network approach using the classical gradient descent method and the population algorithm of gravitational search is investigated in the framework of this work.

B. Gradient descent algorithm

The gradient descent method is based on finding the extremum point of the function in the direction to the negative of the gradient, which converges quadratically and demonstrates high performance. This method, when applied to functions of the quadratic type in the field of real numbers, demonstrates good convergence.

Gradient descent algorithm:

1. The initial approximation y_0, calculation error ε, multiplier of the gradient descent of α and the number of steps of the algorithm $k = 0$.
2. Calculate the initial direction of the anti-gradient:
$$j = 0, \quad S_k^j = -\nabla E(y_k), \quad \mathrm{y}_k^j = y_k. \quad (23)$$
3. Calculate the following approximation:
$$\mathrm{y}_k^{j+1} = y_k^j + \alpha S_k^j, \quad S_k^{j+1} = -\nabla E(\mathrm{y}_k^{j+1}). \quad (24)$$
If $\| S_k^{j+1} \| < \varepsilon$ or $\| \mathrm{y}_k^{j+1} - \mathrm{y}_k^j \| < \varepsilon$, then $\bar{y} = y_k^{j+1}$, end.
Otherwise: If $(j+1) < n$, then $j = j + 1$ and to step 3, otherwise - $\mathrm{y}_{k+1}^j = \mathrm{y}_k^j, k = k + 1$ and to step 2.

C. Gravitational search algorithm

The gravitational search algorithm (GS) appeared relatively recently (2009) and was a logical development of the Central force method. GS is based on the laws of gravity and the interaction of masses. In practice, the method works more accurately than genetic algorithms with real coding and classic PSO.

General scheme of the algorithm GS:

1. Random generation of the system (population is a set of different pairs of weight coefficients)
$$S = \{p_i^1, p_i^2, ..., p_i^3\}_{i=1}^N, \quad (25)$$
where N is the maximum number of particles in the system.
2. Determination of the fitness function $f(p_i)$ (minimized function) of each particle.
3. Updating the values of the gravitational constant, best and worst particles, as well as masses.

In the simplest case, all three weight (passive, active, inertial) are equal: $M_{ai}(t) = M_{pi}(t) = M_{ii}(t) = M_i$. Then the mass value can be recalculated by the formula:

$$\begin{aligned} M_i(t) &= \frac{m_i(t)}{\sum_{j=1}^{N} m_j(t)}, \\ m_i(t) &= \frac{f(p_i) - \max\limits_{j=1,N} f(p_j)}{\min\limits_{j=1,N} f(p_j) - \max\limits_{j=1,N} f(p_j)}. \end{aligned} \quad (26)$$

The value of the gravitational constant should be determined monotonically decreasing function, depending on the initial value of the constant G_0 and the time t, for example:

$$G(t) = \frac{G_0}{e^{\beta t}}, \beta > 0. \quad (27)$$

4. Calculation of the resultant force in different directions:
$$\begin{aligned} F_i(t) &= \sum_{j=1, j \neq i}^{N} \xi_i F_{ij}(t), \\ F_{ij}(t) &= G(t) \frac{M_{pi}(t) \times M_{aj}(t)}{\| p_i, p_j \| + \varepsilon} (p_j(t) - p_i(t)), \end{aligned} \quad (28)$$
where ξ_i -random variables uniformly distributed from zero to one; M_{aj} — active gravitational mass of the j-th particle; M_{pi} — passive gravitational mass of the i-th particle; ε — small constant.
5. Calculation of accelerations and speeds:

$$a_i(t) = M_{ii}(t),$$
$$v_i(t+1) = (\varsigma_1, .., \varsigma_n)^T \times v_i(t) + \frac{F_i(t)}{a_i(t)}, \quad (29)$$

where ς_k is a random variable uniformly distributed from zero to one, M_{ii} is the inert mass of the i-th particle.

6. Update the positions of particles (pairs of weighting coefficients)

$$p_i(t+1) = p_i(t) + v_i(t+1). \quad (30)$$

7. Repeat steps 2 through 6 until the completion criterion is met (for example, achieving a given accuracy).

IV. Comparative Analysis Of Convergence Of Population And Gradient Algorithms For Optimization Of Neural Network Solution Of Optimal Control Problems

We investigate the convergence of the neural network approach to solving the optimal control problem by comparing the error of approximation schemes that use different optimization algorithms at the stage of updating the weight coefficients: the gradient descent method and the gravitational search.

We consider particular examples of optimal control problems that have an analytical solution.

Example A

Consider the problem of optimal control of the form:

$$\begin{cases} \int_0^1 u^2(t)dt - x(1) \to \min \\ \dot{x}(t) = x(t) + u(t) \\ x(0) = 0 \end{cases} \quad (31)$$

Analytical solution of the problem (31) has the form:

$$x^*(t) = -\frac{e^{1-t}}{2} + \frac{e^{1+t}}{2}, u^*(t) = \frac{e^{1-t}}{2}.$$

The solution of the optimal control problem, according to (15), will be sought in the following form:

$$\begin{cases} \tilde{x}(t) = t \cdot n_x \\ \tilde{\lambda}(t) = 1 + (t-1) \cdot n_\lambda \\ \tilde{u}(t) = n_u \end{cases} \quad (32)$$

The functions n_x, n_λ, n_u are defined according to (14).

The Lagrange function for the problem (21) has the form:

$$L(t, x, u, \lambda) = u^2(t) + \lambda(x(t) + u(t) - \dot{x}(t)) \quad (33)$$

The trial solutions (32) are a universal approximation and must satisfy the conditions (6) - (10). Thus have

$$\dot{x}(t) - x(t) - u(t) = 0, \quad (34)$$

$$\dot{\lambda} + \lambda = 0, \quad (35)$$

$$\lambda - 2u = 0. \quad (36)$$

In order to reformulate (31) into a nonlinear optimization problem, first fix the system (34) - (36) at the points $t_k, k = 1, .., r$ of the interval $[t_0 = 0, t_f = 1]$ and then define the optimization problem as

$$\min_y E(y) = \frac{1}{2}\sum_{k=1}^{r}\{E_1(t_k, y) + E_2(t_k, y) + E_3(t_k, y)\}, \quad (37)$$

where $y = (w_x, w_\lambda, w_u, b_x, b_\lambda, b_u, v_x, v_\lambda, v_u)^T$ and

$$\begin{cases} E_1(t_k, y) = [\dot{x}_k - x_k - u_k]^2, \\ E_2(t_k, y) = [\dot{\lambda}_k + \lambda_k]^2, \quad k = 1, ..., r \\ E_3(t_k, y) = [\lambda_k - 2u_k]^2, \end{cases} \quad (38)$$

We consider the results of the neural network approach with different algorithms for optimization of weight coefficients.

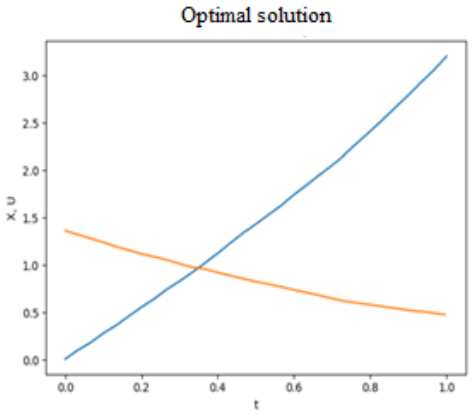

Fig. 1. Neural network solution with gradient descent method ($N = 5$ $E(y) \approx 0,0991; count = 4169$)

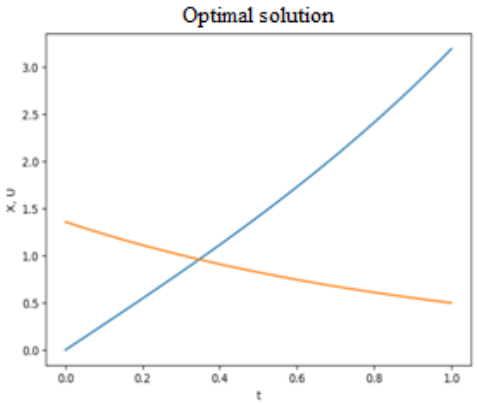

Fig. 2. Neural network solution with gravitational search ($N = 5$; $E(y) \approx 0,0986; count = 1871$)

The figures above show graphs of optimal control and the corresponding optimal trajectory for the number of neurons N=5 of the neural network algorithm using the gradient descent method (Fig.1) and gravitational search (Fig. 2). The

population size for the gravitational search algorithm is $size = 500$.

The General nature of the optimal trajectories is similar, which indicates the adequacy of the considered approximation model. The errors of the algorithms are comparable and have first order of accuracy: $E(y) = 0,099152964$ and $E(y) = 0,098693539$ respectively. The number of iterations of the gravitational search $count = 1871$ is much less than the gravitational descent $count = 4169$. Thus, we can conclude that the gradient search algorithm works more efficiently for this problem and has a better convergence rate.

Example B

Consider the problem of optimal control of the form:

$$\begin{cases} \int_0^1 u^2(t)dt \to \min \\ \dot{x}(t) = u(t) \\ x(0) = e \end{cases} \tag{39}$$

The solution of the optimal control problem, according to (15), will be sought in the following form:

$$\begin{cases} \tilde{x}(t) = e + t \cdot n_x \\ \tilde{\lambda}(t) = t \cdot n_\lambda \\ \tilde{u}(t) = n_u \end{cases} \tag{40}$$

The functions n_x, n_λ, n_u are defined according to (14).

The Lagrange function for the problem (39) has the form:

$$L(t, x, u, \lambda) = u^2(t) + \lambda(u(t) - \dot{x}(t)) \tag{41}$$

The trial solutions (40) are a universal approximation and must satisfy the conditions (6) - (10). Thus have

$$\dot{x}(t) - u(t) = 0, \tag{42}$$

$$\dot{\lambda} = 0, \tag{43}$$

$$\lambda - 2u = 0. \tag{44}$$

In order to reformulate (39) into a nonlinear optimization problem, first fix the system (42) - (44) at the points $t_k, k = 1,..,r$ of the interval $[t_0 = 0, t_f = 1]$ and then define the optimization problem as

$$\min_y E(y) = \frac{1}{2}\sum_{k=1}^{r}\{E_1(t_k, y) + \\ + E_2(t_k, y) + E_3(t_k, y)\}, \tag{45}$$

where $y = (w_x, w_\lambda, w_u, b_x, b_\lambda, b_u, v_x, v_\lambda, v_u)^T$ and

$$\begin{cases} E_1(t_k, y) = [\dot{x}_k - u_k]^2, \\ E_2(t_k, y) = [\dot{\lambda}_k]^2, \quad k = 1,...,r \\ E_3(t_k, y) = [\lambda_k - 2u_k]^2, \end{cases} \tag{46}$$

We consider the results of the neural network approach with different algorithms for optimization of weight coefficients for the problem (39).

The figures above show graphs of optimal control and the corresponding optimal trajectory for the number of neurons N=5 of the neural network algorithm using the gradient descent method (Fig. 3) and gravitational search (Fig. 4). The population size for the gravitational search algorithm is $size = 500$.

The General nature of the optimal trajectories is similar, which indicates the adequacy of the considered approximation model. The errors of the algorithms are comparable and have first order of accuracy: $E(y) = 0,098956827$ and $E(y) = 0,099545928$ respectively. The number of iterations of the gravitational search $count = 259$ is less than the gravitational descent $count = 274$.

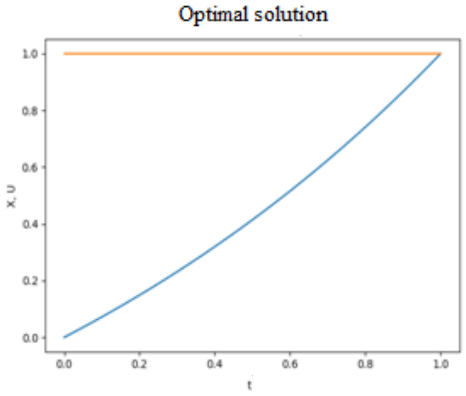

Fig. 3. Neural network solution with gradient descent method ($N = 5$; $E(y) \approx 0,0989$; $count = 274$)

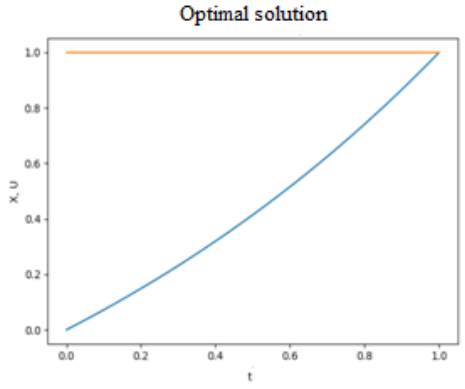

Fig. 4. Neural network solution with gravitational search ($N = 5$; $E(y) \approx 0,0995$; $count = 259$)

Thus, we can conclude that the gradient search algorithm works more efficiently for this problem, but the gap in the number of iterations has decreased relative to example A. This behavior may be due to the gully of the minimized Lagrange function.

V. Conclusion

In this paper, we study the functional representation of the optimal control problem solution without restrictions using a neural network approach. Based on the necessary first-order optimality conditions, the original problem is reduced to a nonlinear optimization problem where the

weights and displacements associated with all neurons are unknown.

The study considers the modification of the neural network approach to solving the optimal control problem at the stage of updating the weight coefficients. The minimization of the error function of the neural network solution is carried out by the gradient descent method, as well as by the population gravity search algorithm. The examples demonstrating the effectiveness of the considered methods are considered. A comparative analysis of the convergence of the algorithms used showed that the gravitational search algorithm, which requires the least number of iterations to achieve accuracy, is more efficient.

ACKNOWLEDGMENT

The researches were carried out in accordance with the plan of research work in 2019-2020 years of FSSI «Federal Research Centre of Biological Systems and Agro-technologies of the RAS» (No. 0761-2019-0004), and was funded by the President of the Russian Federation within the grant for state support of young Russian scientists (MK-860.2019.9), RFBR, according to the research project No. 18-37-00400, as well as the Ministry of education of the Orenburg region in the framework of research "Development of intelligent virtual assistant for planning trips to the sights of the Orenburg region".

REFERENCES

[1] Bolodurina I.P.., Parfenov D.I., Antsyfirova L.M. “Solution of discontinuous problem of optimal control over the individual human capital development by the numerical method”, vol. 51, *IFAC-PapersOnLine*, 2018, pp. 19-24.

[2] Qianxiao L., Shuji H. “An Optimal Control Approach to Deep Learning and Applications to Discrete-Weight Neural Networks”, *ICML*, 2018, pp. 1-10.

[3] Buskensa C., Maurerb H. “SQP-methods for solving optimal control problems with control and state constraints: adjoint variables, sensitivity analysis and real-time control”, Vol. 120, *Journal of Computational and Applied Mathematics*, 2000, pp.85-108.

[4] Abu-Khalaf M., Lewis F.L., “Nearly Optimal State Feedback Control of Constrained Nonlinear Systems Using a Neural Networks HJB Approach”. Vol. 28, *IFAC Annual Reviews in Control*, 2004, pp. 239-251.

[5] Jajarmi A, Pariz N, Vahidian A, Effati S “A novel modal series representation approach to solve a class of nonlinear optimal control problems”, *Int J Innov Comput Inf Control*, 2011, 7:1413–1425.

[6] Becerikli Y., Konarm A.F., Samad T. “Intelligent optimal control with dynamic neural networks”, *Neural Netw*, 2003, 16:251–259.

[7] Hunt K.J., Sbarbaro D., Zbikowski R. “Neural ntewirks for control system – a survey ”, Vol. 28, *Automatika*, 1992, pp.1083-1112.

[8] Nazemi A.R., Shabani M.M. “Numerical solution of the time-delayed optimal control problems with hybrid functions”, *IMA J Math*, 2015, 32: 623–638.

[9] Suykeyns J.A.K., Bersini H. “Neural Control Thery: an Overview”, *Model based Information Processing Systems*, 2017, pp.1-15

[10] Nazemi A., Karami R, “A Neural Network Approach for Solving Optimal Control Problems with Inequality Constraints and Some Applications”, *Neural Process Lett*, 2016, pp.1-29