On the Convergence of an Efficient and Robust Dynamic Neural Network Concept with Application to Solving Traveling Salesman Problems

ELNUR NOROV[1], SHAKHZOD TASHMETOV[1], KHABIBULLO NOSIROV[1],
MAKHIRAKHON RAKHMATULLAEVA[1], AHMED YUSUPOV[1],
JEAN CHAMBERLAIN CHEDJOU[2*]
[1]Department of Television and Radio Broadcasting System,
Tashkent University of Information Technologies (TUIT),
Tashkent 100084, Amir Temur Avenue 108,
UZBEKISTAN
[2]Department of Smart Systems Technologies, University of Klagenfurt,
Universitätsstraße 65/67, 9020 Klagenfurt,
AUSTRIA
**Corresponding author*

Abstract: - In our previous contributions [20, 21, 22], we have clearly demonstrated that the dynamic neural network concept (DNN-concept) for solving shortest path problems (SPP) and traveling salesman problems (TSP) outperforms the best heuristic methods proposed by the literature. However, in our numerous contributions and also according to the literature, the effects of the step sizes of both "decision neurons" and "multiplier neurons" on the convergence properties of the "DNN-concept" are still not investigated. The aim of our contribution is to enrich the literature by investigating, for the first time, the convergence properties of the DNN-concept for solving traveling salesman problems. We develop a mathematical model for the efficient and robust solving the traveling salesman problem (TSP). Based on the numerical study, the convergence properties of the model developed (i.e., the DNN-concept for solving TSP) is investigated. Ranges (or windows) of variation of the parameters of the developed mathematical model are determined (identified) to ensure (guarantee) the detection of the exact TSP solution/tour. In order to validate the mathematical model developed for solving TSP, a bifurcation analysis is carried out using the developed mathematical model. Various bifurcation diagrams are obtained numerically. The bifurcation diagrams obtained reveal the ranges of variation of some key parameters of the model developed to ensure (or guarantee) the convergence of the DNN-concept to the exact TSP-solution (i.e., global minimum). Concrete examples of graphs are considered and various numerical simulations are performed as proof of concept. Finally, a comparison of the results obtained with the results published in [17]-[18] lead to a very good agreement.

Key-Words: - Dynamic Neural Network, Bifurcation analysis, Convergence to exact TSP solutions.

Received: April 12, 2022. Revised: October 27, 2022. Accepted: November 21, 2022. Published: December 31, 2022.

1 Introduction

During the past couple of decades the traveling salesman problem (TSP) has been studied in many fields of science and engineering, leading to a great variety of applications such as: ordering-picking in a warehouse [1], hot rolling scheduling [2], ship scheduling [3], pickup and delivery [4], carrier-vehicle systems [5], information processing [6], optimization of drone-assisted parcel delivery [7], optimization of traffic factors [8], data clustering [9], X-ray crystallography [10], printed circuit board production [11], in-vehicle route guidance [12], automated guided vehicle systems [13], multipath traffic assignment [14], traffic light control [15], real-time traffic information [16], just to name a few. These applications witness the tremendous attention devoted during the past couple of decades to the development of new methods and algorithms for solving TSP. However most of the traditional methods and algorithms so far developed are generally prone to limitations/drawbacks described in [20, 21, 22]. These limitations/drawbacks do justify the focus (in this paper) on the development of a new concept for solving TSP. Some well known limitations of traditional methods and algorithms are: the weak robustness, low accuracy, weak stability, poor flexibility, low scalability potential,

and failures to converge to the exact/true TSP solution/tour. One can refer to [20, 21, 22] for a full description of the drawbacks/limitations of the heuristic methods. The new DNN solver concept developed (in this paper) is a systematic analytical framework, which does efficiently satisfy all key performance metrics (i.e., robustness, accuracy, flexibility, scalability, convergence to the exact TSP solution, subtours avoidance in TSP, etc.). A further limitation of the traditional methods/concepts/algorithms for solving TSP is the fact that they do not provide a systematic analytical framework, which could be used to handle or overcome the aforementioned limitations.
Further significant studies on the topic addressed in this paper can be found in [24-25]. These references provide a theoretical framework that might help understanding the modeling framework developed in this paper.

This paper is a contribution to the development of a systematic analytical concept which efficiently addresses and overcomes the aforementioned limitations (drawbacks) of the traditional concepts, methods, and algorithms for solving TSP. The main focus is on the mathematical modeling of the TSP problem and the analysis of the convergence of the resulting differential equations obtained as the mathematical model of the DNN-concept for solving TSP problems. A mathematical theory is developed to guarantee the convergence of the DNN- concept to the exact traveling salesman solution. The bifurcation analysis is further carried out in order to depict (or determine) the ranges of the control-parameters in which the convergence to the exact TSP-solutions is always ensured. Many bifurcation diagrams are obtained and depicted (i.e., represented) in order to confirm the convergence of the DNN concept to exact TSP solutions.

The paper is organized as follows. Section 2 focuses on the mathematical modeling of the TSP problem. A full description of the modeling methodology leading to the derivation of the mathematical model of the novel TSP solver-system (i.e., the DNN concept for solving TSP) is given. The mathematical model obtained is expressed in the form of coupled ordinary differential equations (ODE). Section 3 is focused on the numerical investigation of the convergence of the mathematical model developed to the exact TSP solution. The numerical study is mainly focused on the bifurcation analysis, aiming at determining and depicting the suitable values and/or ranges/windows of the parameters (or coefficients) of the mathematical model of the DNN concept for which the convergence to the exact TSP solution is ensured (guaranteed). Various bifurcation diagrams are obtained in terms of the control-parameters of bifurcation, which are the step sizes of both decision neurons and multiplier neurons. The resulting bifurcation diagrams (obtained numerically) clearly reveal the effects of the decision and multiplier neurons on the convergence of the DNN concept to the exact TSP solution. Section 4 is devoted to concluding remarks and outlook. In the outlook, several open research questions of interest for further investigations are briefly formulated.

2 Mathematical Modeling of the TSP

In subsections below we present the full details of the procedure leading to the derivation of the mathematical model of the DNN-concept for solving TSP (expressed into the form of coupled ordinary differential equations ODEs).

2.1 Objective function

The TSP in any given graph network G is formulated by the objective function $\Im$ expressed in (1). The function $f(\vec{x})$ represents the total cost corresponding to the full size of G. The integers j and k are the indexes of nodes in G and, M is the magnitude of G.

$$\Im = Min\left[f(\vec{x}) = \sum_{\substack{j=1 \\ j\neq k}}^{M} \sum_{\substack{k=1 \\ k\neq j}}^{M} \omega_{k\to j} x_{k\to j}\right] \qquad (1)$$

The generalized form of (1) is given by (2). The symbol "°" denotes the elementwise product (called Hadamard-product).

$$\Im = Min\left[f(\vec{x}) = \sum_{p=1}^{N} \omega_p \circ x_p\right] \qquad (2)$$

where $x_p = [x_1, \ldots, x_N]^T$ and $\omega_p = [\,\omega_1, \ldots, \omega_N]^T$ are respectively the states and weights of all edges of G (N is the size of G). The components of x_p are binary numbers and thus x_p is a binary vector. The components of x_p equal to "1" reveal the edges belonging to the TSP solution/tour, and components of x_p equal to "0" stand for those edges which do not belong to the TSP solution/tour.

2.2 Constraints

The objective function in (2) is subject to a series constraints, which are formulated to fulfill all key requirements related to TSP.

In our recent contribution (see [20]) we have defined the general condition for assigning attributes to nodes by (3). The notation $k \leftarrow j$ stands for all

incoming edges to a node with index k. Similarly, the notation $k \to j$ stands for all outgoing edges from a node with index k.

$$\sum_{\substack{j=1 \\ j\neq k}}^{M} x_{k\to j} - \sum_{\substack{j=1 \\ j\neq k}}^{M} x_{k\leftarrow j} = \delta \qquad (3)$$

2.2.1 Constraint 1

This constraint insures the belonging of each node of G to the TSP solution. To fulfil (or achieve) this constraint, each node of G is assigned the attribute of intermediate node by choosing $\delta = 0$ in (3) [20]. Thus, applying (3) to all nodes of G (under the condition $\delta = 0$) leads to the general form (4).

$$g_1(x_p) = Ax_p = 0 \qquad (4)$$

The symbol A in (4) stands for the incidence matrix of G.

Remark 1. For the sake of providing a clear understanding of some fundamental issues like: a) the modeling procedure, b) the constraints formulated, c) the mathematical models obtained, and d) the parameters of the mathematical models obtained, we will be using (throughout the paper), for illustrative purposes, the example of the six-city graph network published in references [17]-[18] (see Fig. 1).

The elements/values in rows 1 to 6 and columns 1 to 30 of the matrix A (see below) correspond to the specific case of Fig. 1. In row 1 of A the values "1" reveal the connection to P_1 of the outgoing edges (from P_1) denoted x_1, x_3, x_6, x_8, and x_9. The values "-1" reveal the connection to P_1 of the incoming edges (to P_1) denoted x_2, x_4, x_5, x_7, and x_{10}. Similarly, the columns 11 and 12 of the matrix A reveal the connection of P_2 and P_3 by the pair of parallel edges (x_{11}, x_{12}).

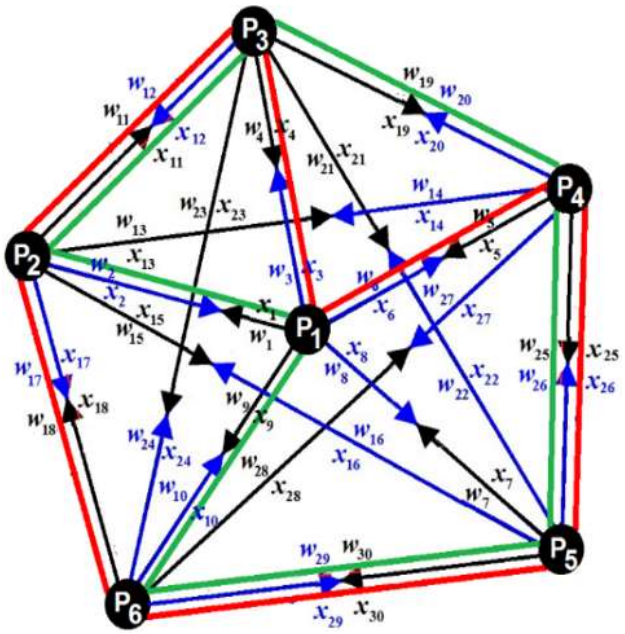


Fig. 1: Topology of a six-city graph published in [17]-[18]. This is a completed bidirectional graph of magnitude 6 and size 30. The states of edges are denoted by $\boldsymbol{x_i}$ and the weights (i.e. edges costs) are $\boldsymbol{\omega_i}$. Examples of TSP tours are displayed in green color (the TSP solution proposed in [17]) and red color (the TSP solution proposed in [18]).

Remark 2. The procedure (described above) to determine the incidence matrix A is applicable to any graph network G regardless of the topology, size and magnitude of the graph.

$$A = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & x_{10} & x_{11} & x_{12} & x_{13} & x_{14} & x_{15} & x_{16} \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 1 & -1 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \square & \square & \square & \square & \square & \square & \square & \square & \square & \square & \square & \square & \square & \square & \square & \square \\ \vdots & & \ddots & \square & \ddots & \square & \ddots & \square & \ddots & \square & \ddots & \square & \ddots & \square & \ddots & \square \\ \square & \cdots & \square & \cdots & \square & \cdots & \square & \cdots & \square & \cdots & \square & \cdots & \square & \cdots & \square & \cdots \end{bmatrix}$$

$$\begin{bmatrix} x_{17} & x_{18} & x_{19} & x_{20} & x_{21} & x_{22} & x_{23} & x_{24} & x_{25} & x_{26} & x_{27} & x_{28} & \cdots & x_N & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & \square & P_1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & \square & P_2 \\ 0 & 0 & 1 & -1 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 & \cdots & \square & P_3 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & \cdots & \square & P_4 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & -1 & 1 & 0 & 0 & \cdots & \square & P_5 \\ -1 & 1 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & -1 & 1 & \cdots & \square & P_6 \\ \square & \square & \square & \square & \square & \square & \square & \square & \square & \square & \square & \square & \cdots & \square & \vdots \\ \ddots & \square & \ddots & \square & \ddots & \square & \ddots & \square & \ddots & \square & \ddots & \square & \ddots & \square & P_M \\ \square & \cdots & \square & \cdots & \square & \cdots & \square & \cdots & \square & \cdots & \square & \cdots & \square & \cdots & \end{bmatrix}$$

2.2.2 Constraint 2

This constraint ensures the connection of each node of G by exactly "one incoming edge" and "one outgoing edge". This constraint is modeled by (5), assuming (4) is satisfied. Thus the constraints 1 and 2 are complementary [20].

$$\sum_{\substack{j=1 \\ j\neq k}}^{M} x_{k\to j} + \sum_{\substack{j=1 \\ j\neq k}}^{M} x_{k\leftarrow j} = 2 \qquad (5)$$

Applying (5) to all nodes of G leads to the general form (6).

$$g_2(x_p) = \left(|A|x_p - 2\right) = 0 \qquad (6)$$

The quantity $|A|$ in (6) denotes the absolute value of A.

2.2.3 Constraint 3

This constraint is formulated to avoid the simultaneous belonging of parallel edges (x_{2n-1}, x_{2n}) to the TSP solution/tour. This constraint is modeled by applying (7) to each pair of parallel edges. Let's mention that parallel edges are edges expressing the bidirectional communication between each pair of nodes.

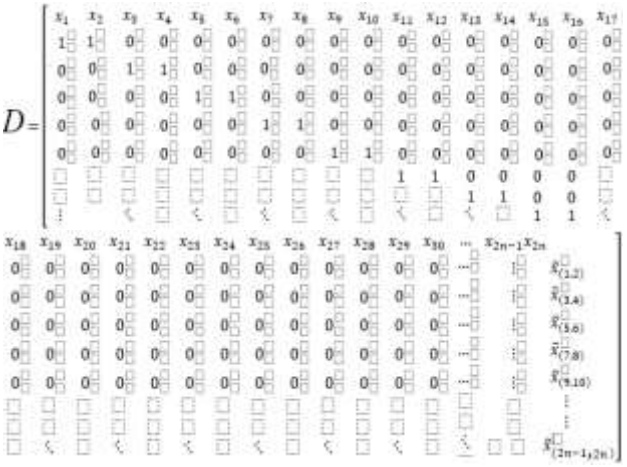
$$(x_{2n-1} + x_{2n})(x_{2n-1} + x_{2n} - 1) = 0 \qquad (7)$$

According to (7) each pair of parallel edges can take only one of the states $(x_{2n-1}, x_{2n}) \in \{(0,0),(0,1),(1,0)\}$; thus the case $(x_{2n-1}, x_{2n}) = (1,1)$ which corresponds to parallel edges is surely avoided under condition (7). An illustrative example of applying (7) to parallel edges $(x_1, x_2) = \tilde{x}_{(1,2)}$, $(x_3, x_4) = \tilde{x}_{(3,4)}$,…,$(x_{29}, x_{30}) = \tilde{x}_{(29,30)}$ in Fig.1 is as follows: $(x_1 + x_2)(x_1 + x_2 - 1) = 0$, $(x_3 + x_4)(x_3 + x_4 - 1) = 0$,..,$(x_{29} + x_{30})(x_{29} + x_{30} - 1) = 0$. Applying (7) to all parallel edges of G denoted by $(x_{2n-1}, x_{2n}) = \tilde{x}_{(2n-1,2n)}$ leads to the general expression (8).

$$g_3(x_p) = Dx_p{}^{\circ}\left(Dx_p - 1\right) = 0 \qquad (8)$$

The matrix D is of size $(N/2) \times N$ and D expresses the connectivity of all pairs of parallel edges $(x_{2n-1}, x_{2n}) = \tilde{x}_{(2n-1,2n)}$ in the graph G. Let's mention here that, in the general form, the last pair of parallel edges in the graph G is obtained for $n = \frac{N}{2}$ and this corresponds to $\tilde{x}_{(N-1,N)}$.

$D =$



Remark 3. In the matrix D, only the pairs of parallel edges denoted by $\tilde{x}_{(1,2)}$, $\tilde{x}_{(3,4)}$, $\tilde{x}_{(5,6)}$, $\tilde{x}_{(7,8)}$, and $\tilde{x}_{(9,10)}$ are represented. These pairs correspond respectively to (x_1, x_2), (x_3, x_4), (x_5, x_6), (x_7, x_8), and (x_9, x_{10}) in Fig. 1. One can deduce D for the full size of Fig. 1. Note that the form of D is standard. Thus, D holds for any graph G regardless of the topology, size, and magnitude.

Remark 4. Note that the constraint formulated in (8) holds even for a numbering of edges which is different from the numbering chosen in the example of Fig. 1; in case the numbering differs from that currently used in Fig. 1, one obtains a matrix D which is not a diagonal matrix, but which still satisfies the constraint formulated in (8). For the sake of generalization and to facilitate programming or algorithmic-coding (in case we may consider a complex graph network, e.g., graph of huge size and fully connected), expressing the matrix D into diagonal form is convenient (highly recommended).

2.2.4 Constraint 4

This constraint is formulated to avoid subtours in the TSP solution. Thus the condition (9) is used to ensure (or guarantee) only TSP solutions/tours with single- cycles.

$$\prod_{i=0}^{(\aleph-1)} \left(\sum_{\substack{j=1 \\ x_j \epsilon S_k}}^{\aleph} x_j - i \right) = 0 \qquad (9)$$

In (9), x_j stands for edges belonging to a given subtour S_k, and $\aleph$ is the total number of edges in the subtour S_k.

The expression (9) is applicable to any graph G of known topology. The knowledge of the graph topology leads to the straightforward determination of the sizes of subtours in G. Indeed the subtours of G are obtained according to (9) when the

characteristic relation $2 < \aleph < |V| - 2$ is fulfilled. The quantity $|V|$ stands for the magnitude of G.
In [19] the inequality $\sum_{\substack{e=(u,v)\in E \\ u,v\in S}} x_e \leq |S| - 1$ is proposed for the subtour elimination constraints (SEC), where S is a non-empty subset of the set of nodes of graph G and x_e is a binary variable revealing the state of edges- connectivity. The advantage of (9) when compared to the SEC in [19] is that (9) is used both for the determination of all possible subtours and for the elimination of the subtours. Further advantage is that the equality constraint (9) is appropriate for the modeling of our Neuro-Processor solver while in contrast the inequality constraints SEC in [19] is not appropriate for the mathematical modeling of TSP by ordinary differential equations.

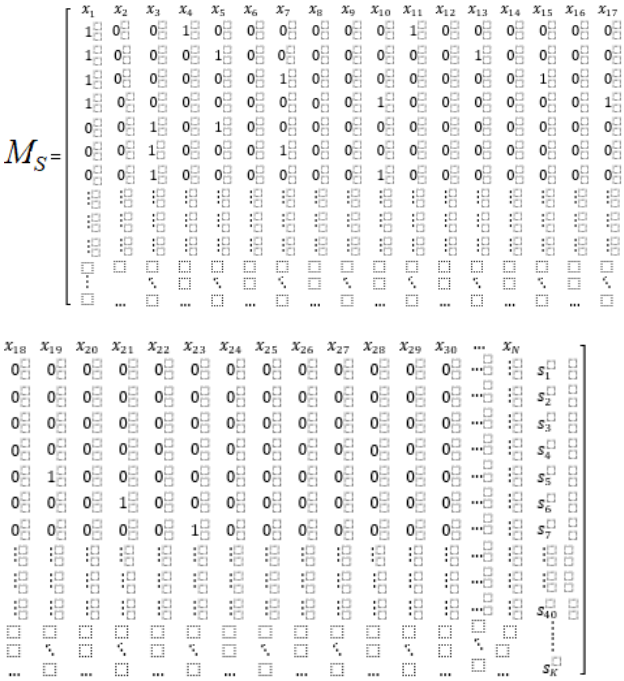
We now consider Fig. 1 to illustrate the application of (9). Let us mention that in the specific case of Fig. 1, $\aleph = 3$ for all subtours S_k. Further, a total of 20 possible TSP solutions (each of which is made-up of a pair of subtours) can be found in Fig. 1. The TSP solutions with two cycles (called multi-tours TSP) are surely avoided when the condition (9) is satisfied. Amongst the TSP solutions with two cycles (in Fig. 1), let us mention, just for illustration, the pair of subtours S_1 made-up of edges (x_1, x_{11}, x_4) and S_1' with edges (x_{26}, x_{27}, x_{29}). Another TSP solution with two cycles (in Fig. 1) is S_2 with edges (x_6, x_{25}, x_7) and S_2' with (x_{11}, x_{23}, x_{18}).

Remark 5. A total of 40 subtours corresponding to 20 TSP solutions (with two cycles each) co-exist in Fig. 1 with several other TSP solutions with single-cycles. Thus, (9) is applied to eliminate subtours and thus ensure only TSP solutions with single-cycles in a given graph network with known topology.

For a given graph G, the subtours S_k are expressed through the matrix M_S. Worth mentioning is that M_S can be easily determined for any given graph network of known topology.

We now consider Fig. 1 as a particular/specific example to illustrate the procedure leading to the determination of M_S. The subtour S_1 (see first row of M_S) is made-up of edges (x_1, x_4, x_{11}). According to (9) the subtour S_1 is avoided by the following analytic equation: $(x_1 + x_4 + x_{11})(x_1 + x_4 + x_{11} - 1)(x_1 + x_4 + x_{11} - 2) = 0$.
It clearly appears that this equation does not admit as solution the case $x_1 = x_4 = x_{11} = 1$ (corresponding to S_1)

$M_S =$



Overall, for any given graph network, the application of (9) to all subtours of the graph leads to the general form (10).

$$g_4(x_p) = M_S x_p \circ (M_S x_p - 1) \circ \ldots \circ (M_S x_p - \aleph + 1) = 0 \quad (10)$$

where $x_p = [x_1, \ldots, x_N]^T$ is the state vector of the edges in G.
The symbol "∘" denotes the elementwise product. The constraint modeled by (10) is used to eliminate the subtours S_P described by the matrix M_S. M_S is of size $\mathbb{N} \times N$ and can be obtained for a given graph network G. N is the size of G and $\mathbb{N}$ is the total number of subtours in G. Note that $\mathbb{N} = 40$ in the specific case of Fig. 1. Also note that only 7 subtours are represented in M_S. The full matrix M_S corresponding to Fig. 1 can be deduced by representing all the 40 subtours. As already defined above, $\aleph$ stands for the total number of edges in a given subtour S_k. The range of variation of $\aleph$ is $2 < \aleph < |V| - 2$, where $|V|$ stands for the magnitude of G. In the particular case of Fig. 1, we have $\aleph = 3$ as all the 40 subtours identified are each with three edges.

2.2.5 Constraint 5

This constraint expressed into the general form (11) ensures the binarization of all edges of G [20].

$$g_5(x_p) = x_p(x_p - 1) = 0 \quad (11)$$

where $x_p = [x_1, \ldots, x_N]^T$ is the state vector of edges in G.

2.2.6 Constraint 6

This constraint improves the robustness of the binarization constraint (11). According to [20], the improvement of the robustness of the convergence properties of all edges of G to binary variables (i.e. "1" and/or "0") is ensured by the augmented Lagrange method. This method is derived from (11) by introducing an additive penalty force expressed in the form of quadratic energy. This justifies the constraint expressed in the general form (12). This constraint was also reported in our references [20].

$$g_6(x_p) = \left[g_5(x_p)\right]^2 = \left(x_p^4 - 2x_p^3 + x_p^2\right) = 0 \quad (12)$$

The next section is concerned with the formulation of the Lagrange function as the total energy of the system (see our Refs. [20], [21], [22], [23]).

2.3 The Basic Differential Multiplier Method

We now transform the constrained optimization problem into an unconstrained optimization problem. This is done by expressing the Lagrange function as a combination of the objective function with the formulated constraints. The overall procedure consists of introducing new vectors of multiplier variables λ_1, λ_2, λ_3, λ_4, λ_5, and λ_6 for the six groups of constraints (4), (6), (8), (10), (11) and (12). Combining the objective function (2) with these constraints leads to the Lagrange function denoted by $\mathcal{L}$.

$$\begin{aligned}\mathcal{L} = f(x_p) + \lambda_1 g_1(x_p) + \lambda_2 g_2(x_p) + \lambda_3 g_3(x_p) \\ +\lambda_4 g_4(x_p) + \lambda_5 g_5(x_p) + \lambda_6 g_6(x_p)\end{aligned} \quad (13)$$

Eq. (13) is now used to obtain the model for solving TSP. The Basic Differential Multiplier Method (BDMM) expressed in (14) is used to obtain the set of coupled ordinary differential equations describing the mathematical model for solving TSP. More details on the application of BDMM is also provided in our refs.[20], [21], [22], [23].

$$\begin{cases} \dfrac{dx_p}{dt} = -\alpha \dfrac{\partial \mathcal{L}}{\partial x_p} \\ \dfrac{d\lambda_1}{dt} = \beta_1 \dfrac{\partial \mathcal{L}}{\partial \lambda_1}\ ;\ \dfrac{d\lambda_2}{dt} = \beta_2 \dfrac{\partial \mathcal{L}}{\partial \lambda_2}\ ;\ \dfrac{d\lambda_3}{dt} = \beta_3 \dfrac{\partial \mathcal{L}}{\partial \lambda_3} \\ \dfrac{d\lambda_4}{dt} = \beta_4 \dfrac{\partial \mathcal{L}}{\partial \lambda_4};\ \dfrac{d\lambda_5}{dt} = \beta_5 \dfrac{\partial \mathcal{L}}{\partial \lambda_5}\ ;\ \dfrac{d\lambda_6}{dt} = \beta_6 \dfrac{\partial \mathcal{L}}{\partial \lambda_6} \end{cases} \quad (14)$$

The expression (14) corresponds to a neural network with anti-symmetric connections between the vectors of multiplier neurons (also called multiplier variables) $\boldsymbol{\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6}$ and all components of the vector of decision neurons (also called decision variables) $\boldsymbol{x_p = [x_1, \ldots, x_N]^T}$. The step sizes for updating decision variables and multiplier variables are denoted by $\boldsymbol{\alpha}$ and $\boldsymbol{\beta_i}$ $(\boldsymbol{i = 1, 2, 3, 4, 5, 6})$ respectively.

Using (13), the partial derivatives are calculated in the BDMM model in (14) and, the resulting mathematical model obtained is expressed in (15).

$$\begin{cases} \dfrac{dx_p}{dt} = -\alpha \Big\{ A^T\lambda_1 + |A|^T\lambda_2 + \left(2D^TDx_p - 1\right)^{\circ} D^T\lambda_3 \\ \quad \left[\left(M_s^TM_Sx_p\right)^{\circ}\left(M_s^TM_Sx_p - 1\right)^{\circ} \ldots {}^{\circ} \right. \\ \left(M_s^TM_Sx_p - \aleph + 1\right) \left.\right] \left[\left(M_s^TM_Sx_p\right)^{-1} + \left(M_s^TM_Sx_p - 1\right)^{-1} \right. \\ \quad \left. + \cdots + \left(M_s^TM_Sx_p - \aleph + 1\right)^{-1}\right]^{\circ} M_s^T\lambda_4 \\ \quad + \left(2x_p - 1\right)^{\circ}\lambda_5 + \left(4x_p^3 - 6x_p^2 + 2x_p\right)^{\circ}\lambda_6 + \omega_p \Big\} \\ \dfrac{d\lambda_1}{dt} = \beta_1\left[Ax_p\right] \\ \dfrac{d\lambda_2}{dt} = \beta_2\left[|A|x_p - 2\right] \\ \dfrac{d\lambda_3}{dt} = \beta_3\left[Dx_p{}^{\circ}\left(Dx_p - 1\right)\right] \\ \dfrac{d\lambda_4}{dt} = \beta_4\left[M_Sx_p{}^{\circ}\left(M_Sx_p - 1\right)^{\circ} \ldots {}^{\circ}\left(M_Sx_p - \aleph_p + 1\right)\right] \\ \dfrac{d\lambda_5}{dt} = \beta_5\left[x_p\left(x_p - 1\right)\right] \\ \dfrac{d\lambda_6}{dt} = \beta_6\left[x_p^4 - 2x_p^3 + x_p^2\right] \end{cases} \quad (15)$$

In Eq. (15), the quantities $\boldsymbol{A^T}$, $\boldsymbol{|A|^T}$, $\boldsymbol{D^T}$ and $\boldsymbol{M_s^T}$ denote the transpose of matrices $\boldsymbol{A}$, $\boldsymbol{|A|}$, $\boldsymbol{D}$ and $\boldsymbol{M_s}$, respectively.

3 Numerical Study

The aim of the numerical study is twofold.

- Validating the mathematical model obtained in this work (see (15)) for solving the traveling salesman problem (TSP). This is achieved by comparing the numerical results obtained when solving the TSP in Fig. 1 using Eq. (15) with the results published in refs. [17]-[18].
- Determining the ranges of parameters α (decision variable) and β_i $(i = 1, 2, 3, 4, 5, 6)$ (multiplier variables) for which the mathematical model in Eq. (15) always converges to the exact TSP solution. This is

achieved by carrying out a bifurcation analysis. This analysis consists of varying the control parameters of bifurcation α and β_i $(i = 1, 2,3,4,5,6)$ in order to depict (i.e., identify and detect) the regions in which the mathematical model in Eq. (15) always converges to the exact/true TSP solution.

In order to compare our results with the published results, we use the same values of weights (i.e., costs of edges) proposed in [18] for the six-city graph in Fig. 1: $\omega_1 = \omega_2 = 25$; $\omega_3 = \omega_4 = 25$; $\omega_5 = \omega_6 = 22$; $\omega_7 = \omega_8 = 25$; $\omega_9 = \omega_{10} = 28$; $\omega_{11} = \omega_{12} = 25$; $\omega_{13} = \omega_{14} = 45$; $\omega_{15} = \omega_{16} = 45$; $\omega_{17} = \omega_{18} = 35$; $\omega_{19} = \omega_{20} = 30$; $\omega_{21} = \omega_{22} = 55$; $\omega_{23} = \omega_{24} = 50$; $\omega_{25} = \omega_{26} = 25$; $\omega_{27} = \omega_{28} = 50$; $\omega_{29} = \omega_{30} = 27$.

The numerical simulation of (15) is performed using the following values of parameters: $\alpha = 0.6$; $\beta_1 = \beta_2 = 50$; $\beta_3 = 0.01$; $\beta_4 = \beta_5 = 50$; $\beta_6 = 0.01$. The 4th order Runge-Kutta algorithm is used for the numerical solving of (15) using the step size $h = 0.001$. The results of x_p obtained as numerical solutions of (15) are depicted in Fig.2. As it appears in Fig. 2 the system (15) undergoes a short transient behavior characterized by damped oscillations, followed by the detection of subtours S1, S2, and S3. Finally the system (15) converges to the optimal TSP tour characterized by the following binary values obtained as numerical solutions of (15): $x_4 = x_6 = x_{11} = x_{18} = x_{25} = x_{30} = 1$ (see Fig. 2). The remaining components of x_p are equal zero (see Fig. 2). Thus the trajectory in red color (see Fig.1) corresponds to the "optimal TSP- tour" denoted $P_1 \rightarrow P_4 \rightarrow P_5 \rightarrow P_6 \rightarrow P_2 \rightarrow P_3 \rightarrow P_1$. The total cost of the optimal TSP solution/tour corresponding to the values of x_p is $C_{total} = 159$. These results are confirmed by [18].

We now consider the bifurcation analysis through the numerical solving of (15). The following values are used: $0.25 \leq \alpha \leq 0.4$; $1 \leq \beta_1 \leq 1.25$; $\beta_2 = 1$; $\beta_3 = 1$; $\beta_4 = 1$; $\beta_5 = 1$; $\beta_6 = 0.001$. The results x_p obtained as solution of (15) are used to plot the 3D-bifurcation diagram in Fig. 3. As already mentioned, the components of x_p converge to binary values (see illustrative examples in Fig. 2). These values are used to evaluate the total cost $\mathcal{L}^*$ of the optimal TSP solution using the expression (13).

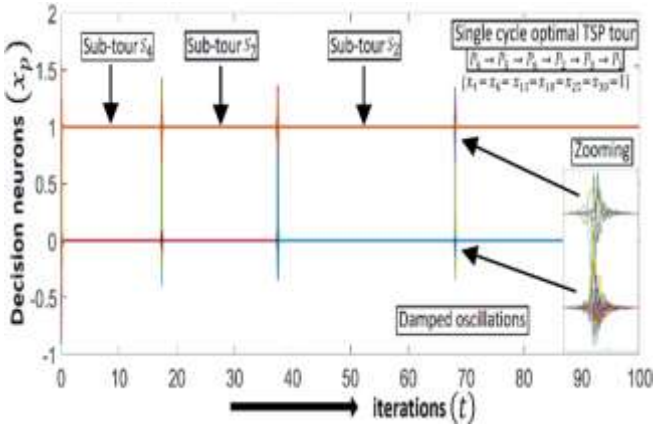


Fig. 2: Results of the numerical solution of the DNN model (15). Damped oscillations occur during the following transitions: $S_4 \rightarrow S_7$; $S_7 \rightarrow S_2$; $S_2 \rightarrow$ *Single cycle optimal TSP tour* (corresponding to the end of the optimization process).

As it appears in Fig. 3 the monitoring (i.e., the variation) of the selected bifurcation parameters in windows/ranges $0.25 \leq \alpha \leq 0.4$ and $1 \leq \beta_1 \leq 1.25$ clearly shows that the optimal TSP- tour (also called exact/true TSP- tour) corresponds to the total cost of $\mathcal{L}^* = 159$. This value depicted in Fig. 3 by blue color cells represents the global minimum point at which the exact/true TSP- tour is obtained. Also, several TSP- tours with the total cost of $\mathcal{L}^* = 160$ are depicted in Fig. 3. The TSP-tours with total costs $\mathcal{L}^* = 160$ correspond to local minima which are very close to the optimal TSP- tour. Therefore, as the chosen (or selected) bifurcation parameters α and β_1 are varied (i.e., monitored), the convergence of the optimization algorithm alternates between the exact/true TSP- solution/tour (global minimum with the total cost of $\mathcal{L}^* = 159$) and several other TSP-tours (corresponding to local minima) with the total costs ($\mathcal{L}^* = 160$) which are very close to the cost of the global minimum.

Worth mentioning is that the alternation of the convergence of the optimization algorithm from the global minimum to several closest local minima (and vice-versa) is a challenging situation commonly faced (or encountered) by the classical optimization algorithms (e.g., traditional neural networks, genetic and heuristic algorithms, etc.). This situation is clearly reported in [18]. Thus, based on the results in Fig. 3, it is clearly demonstrated that using the new mathematical model developed in (15), a suitable choice of the bifurcation parameters α and β_1 can help to overcome the aforementioned challenging situation. For example (see Fig. 3) a suitable choice of the following chosen/selected two bifurcation parameters in the ranges $0.25 \leq \alpha \leq 0.30$ and

$1.05 \leq \beta_1 \leq 1.25$ ensures (or guarantees) the convergence of the mathematical model in (15) to the exact/true TSP- tour corresponding to the global minimum $\mathcal{L}^* = 159$ (see blue cells in Fig. 3). This is a pro (advantage) of the mathematical model in (15) as it is clearly demonstrated (through Fig. 3) that a suitable choice of the parameters of the mathematical model in (15) can help avoiding convergence to local minima and thereby ensuring sure convergence to the unique global minimum corresponding to the exact TSP- solution/tour.

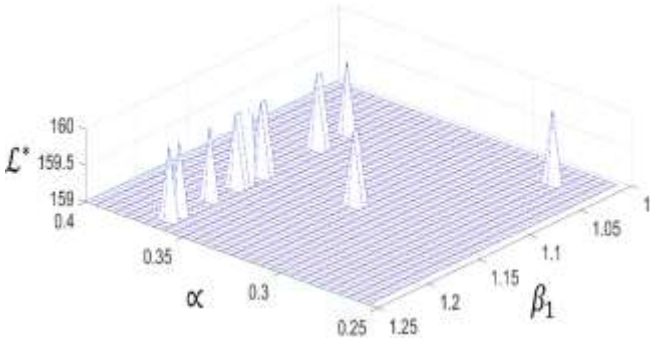


Fig. 3: A 3D-representation of the results of the bifurcation analysis (in terms of α and β_1) obtained using the mathematical model in (15). The total cost of the optimal TSP tour is $\mathcal{L}^* = 159$. This corresponds to the global minimum (Blue cells). The pics (or pulses) correspond to $\mathcal{L}^* = 160$ (total cost of local minima which are very close to the global minimum). The parameters used are: $0.25 \leq \alpha \leq 0.4$; $1 \leq \beta_1 \leq 1.25$; $\beta_2 = 1$; $\beta_3 = 1$; $\beta_4 = 1$; $\beta_5 = 1$; $\beta_6 = 0.001$; $h = 0.05$. One can easily identify from this figure the ranges (or windows) of parameters leading to the sure convergence of the DNN-solver in (15) to the exact TSP solution/tour.

For the sake of benchmarking it should be mentioned that this application example is published in [18] and also that the same optimal TSP-tour denoted by $P_1 \rightarrow P_4 \rightarrow P_5 \rightarrow P_6 \rightarrow P_2 \rightarrow P_3 \rightarrow P_1$ (see red lines in Fig. 1) with the total cost $\mathcal{L}^* = 159$ is obtained. This shows the good agreement between the DNN model developed in this work (see Eq. (15)) and the GA and SA algorithms used in [18].

Similarly to the previous comment (on Fig. 3) regarding the guarantee of convergence of the DNN model in Eq. (15) to the exact TSP solution/tour, it can be found, according to Fig. 4, that the choice of the following chosen/selected bifurcation parameters in the ranges $1 \leq \beta_1 \leq 1.2$ and $1 \leq \beta_2 \leq 1.2$ ensures (or guarantees) the convergence of the DNN model in Eq. (15) to the exact TSP- tour corresponding to the global minimum $\mathcal{L}^* = 159$ (see blue cells in Fig. 4). The pics/pulses in Fig. 4 correspond to TSP- tours representing closest local minima with $\mathcal{L}^* = 160$.

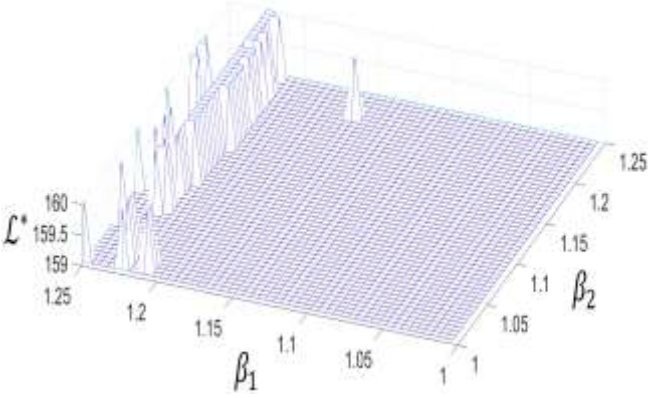


Fig. 4. A 3D-representation of the results of the bifurcation analysis (in terms of β_1 and β_2) obtained using the new DNN model in (15). The total cost of the optimal TSP tour is $\mathcal{L}^* = 159$. This corresponds to the global minimum (Blue cells). Pics (or Pulses) correspond to $\mathcal{L}^* = 160$ (this total cost corresponds to local minima very close to the global minimum). The values of parameters used are: $\alpha = 0.3$; $1 \leq \beta_1 \leq 1.25$; $1 \leq \beta_2 \leq 1.25$; $\beta_3 = 1$; $\beta_4 = 1$; $\beta_5 = 1$; $\beta_6 = 0.001$; $h = 0.05$. One can easily identify from this figure the ranges (or windows) of parameters leading to the sure convergence of the DNN-solver in (15) to the exact TSP solution/tour.

4 Conclusion

This paper has developed a new dynamic neural network (DNN) solver concept for the efficient solving of traveling salesman problems (TSP). The modelling of TSP has been carried out and a mathematical model of the new DNN- solver has been obtained in the form of coupled nonlinear ordinary differential equations.

To validate the DNN-solver model developed in this paper, two application examples published in [17]-[18] have been considered. The DNN model developed has been used to solve the aforementioned application examples. Using the same values of parameters as in [17]-[18], a bifurcation analysis has been carried out numerically based on the mathematical model developed in this work for solving TSP. Using this mathematical model, the ranges/windows of the system parameters have been identified (or detected) under which the mathematical model developed surely and always converges to the exact TSP solution/tour. Finally, a comparison of the TSP solution obtained using the mathematical model developed with the results published in [17]-[18] has led to a very good agreement.

An ongoing work (this for outlook) currently under consideration is the development of a theoretical framework for the analytical investigation of the convergence in order to ensure (or guarantee) the convergence of the new DNN-solver concept developed to the exact TSP solution. The guarantee of convergence has been done in this work numerically. Therefore, it would be very interesting and even challenging to develop a universal and scalable theoretical framework that could help derive and propose the analytical conditions that will ensure (or guarantee) the convergence of the DNN-solver to the exact TSP solution/tour. This could be a significant contribution to the enrichment of the literature as this analysis has not yet been considered by the literature regarding TSP solving.

Acknowledgement:

The first two authors would like to acknowledge the scholarship obtained in the frame of the ERASMUS+ mobility, that has facilitated a six-month research visit in Klagenfurt under the supervision of Prof. Jean Chamberlain Chedjou, during which this work was initiated and completed. The authors of this work are grateful for the valuable comments of the anonymous reviewers, which have greatly improved the quality of this work.

References:

[1] H. D. Ratliff and A. S. Rosenthal, "Order-Picking in a Rectangular Warehouse: A Solvable Case for the Travelling Salesman Problem," *Operations Research,* Vol. 31, 1983, pp. 507-52.

[2] L. Tang, J. Liu, A. Rong and Z. Yang, "A multiple traveling salesman problem model for hot rolling scheduling in Shangai Baoshan Iron and Steel Complex," *European Journal of Operational Research,* Vol. 124, 2000, pp. 267–282.

[3] K. Fagerholt and M. Christiansen, "A travelling salesman problem with allocation, time window and precedence constraints—an application to ship scheduling," *Int. Trans. Oper. Res.,* Vol. 7, 2000, pp. 231–244.

[4] G. Mosheiov, "The traveling salesman problem with pickup and delivery," *Eur. J. Oper. Res.,* Vol. 79, 1994, pp. 299–310.

[5] E. Garone, J.-F. Determe and R. Naldi, "Generalized Traveling Salesman Problem for Carrier-Vehicle Systems," *Journal of Guidance, Control, and Dynamics,* Vol. 37, 2014, pp. 766–774.

[6] X. Kong and C. D. Schunn, "Global vs. local information processing in visual/spatial problem solving: The case of traveling salesman problem," *Cognitive Systems Research,* Vol. 8, 2007, pp. 192–207.

[7] C. C. Murray and A. G. Chu, "The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery," *Transportation Research Part C: Emerging Technologies,* Vol. 54, 2015, pp. 86-109.

[8] M. Mavrovouniotis and S. Yang, "Ant colony optimization with immigrants' schemes for the dynamic travelling salesman problem with traffic factors," *Applied Soft Computing,* Vol. 13, 2013, pp. 4023-4037.

[9] J. K. Lenstra, "Clustering a Data Array and the Traveling-Salesman Problem," *Operations Research,* Vol. 22, 1974, pp. 413–414.

[10] R. E. Bland, and D. E. Shallcross, "Large traveling salesman problem arising from experiments in X-ray crystallography: a preliminary report on computation," *Operations Research Letters,* Vol. 8, 1989, pp. 125-128.

[11] G. Reinelt, "A Case Study: TSPs in Printed Circuit Board Production: The Traveling Salesman," *Lecture Notes in Computer Science book series (LNCS),* Vol. 840, 2001, pp. 187–199.

[12] L. Fu, "An adaptive routing algorithm for in-vehicle route guidance systems with real-time information," *Methodological,* Vol. 35, 2001, pp. 749–765.

[13] R. J. Gaskins and J. M. A. Tanchoco, "Flow path design for automated guided vehicle systems," *International Journal of Production Research,* Vol. 25, 1987, pp. 667–676.

[14] R. B. Dial, "A probabilistic multipath traffic assignment model which obviates path enumeration," *Transportation research,* Vol. 5, 1971, pp. 83–111.

[15] W. Wen, "A dynamic and automatic traffic light control expert system for solving the road congestion problem," *Expert Systems with Applications,* Vol. 34, 2008, pp. 2370-2381.

[16] T. Cheong and C. C. White, III, "Dynamic traveling salesman problem: Value of real-time traffic information," *IEEE Trans. Intell. Transp. Syst.,* Vol. 13, 2012, pp. 619-630.

[17] M. Niendorf, P. T. Kabamba and A. R. Girard, "Stability of Solutions to Classes of Traveling Salesman Problems," *IEEE Trans. Cybern.*, Vol. 46, 2016, pp. 973–985.

[18] J. Li, M.-C. Zhou, Q. Sun, X. Dai and X. Yu, "Colored Traveling Salesman Problem," *IEEE Trans. Cybern.*, Vol. 45, 2015, pp. 2390–2401.

[19] XU. Pferschy and R. Staněk, "Generating subtour elimination constraints for the TSP from pure integer solutions," *Central European J. Oper. Res.*, Vol. 25, 2017, pp. 231–260.

[20] Jean Chamberlain Chedjou, and Kyandoghere Kyamakya, "An efficient, scalable, and robust neuro-processor-based concept for solving single-cycle traveling salesman problems in complex and dynamically reconfigurable graph networks," *IEEE Access*, Vol. 8, 2020, pp. 42297-42324.

[21] Jean Chamberlain Chedjou, and Kyandoghere Kyamakya, "A universal concept for robust solving of shortest path problems in dynamically reconfigurable graphs," *Mathematical Problems in Engineering*, Vol. 2015, 2015, pp. 1-23.

[22] Jean Chamberlain Chedjou, and Kyandoghere Kyamakya, "Benchmarking a recurrent neural network based efficient shortest path problem (SPP) solver concept under difficult dynamic parameter settings conditions," *Neurocomputing Elsevier*, Vol. 196, 2016, pp. 175-209.

[23] J. C. Chedjou and K. Kyamakya, "A universal concept based on cellular neural networks for ultrafast and flexible solving of differential equations," *IEEE Trans. Neural Netw. Learn. Syst.*, Vol. 46, n0. 4, pp. 749-762, April 2015.

[24] Abd Elhakim Lamairia, "Nonexistence Results of Global Solutions for Fractional Order Integral Equations on the Heisenberg Group", WSEAS Transactions on Systems, vol. 21, pp. 382-386, 2022.

[25] Nongluk Viriyapong, "Modification of Sumudu Decomposition Method for Nonlinear Fractional Volterra Integro-Differential Equations", WSEAS Transactions on Mathematics, vol. 21, pp. 187-195, 2022.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

Prof. Jean Chamberlain Chedjou, defined and proposed the research topic as supervisor (external). He also checked the correctness of the mathematical formulas and the wording of the submitted article.

Prof. Ahmed Yusupov, checked the simulation algorithms and the issues related to graph theory.

Prof. Nosirov Khabibullo is the local supervisor. He helped checking the developed algorithms and review debugging.

Prof. Makhirakhon Rakhmatullaeva is a mathematician. She conducted a thorough verification of the correctness of all mathematical formulas. She also proposed the idea how to check the convergence to the unique solution which is the global minimum.

MSc. Tashmetov Shakhzod, contributed to numerical coding by developing the MATLAB codes used for numerical study. He also performed mathematical calculations and contributed to the writing of the article.

MSc. Norov Elnur, contributed to numerical coding by developing the MATLAB codes used for numerical study. He also performed mathematical calculations and contributed to the writing of the article.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

The first two authors would like to acknowledge the scholarship obtained in the frame of the ERASMUS+ mobility, that has facilitated a six-month research visit in Klagenfurt under the supervision of Prof. Jean Chamberlain Chedjou, during which this work was initiated and completed.

Conflict of Interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.