Optimisation of the Fuzzy Model of Nonlinear Systems by Society-inspired Heuristics

Nouri Benaidja
Common trunk, Faculty of technology
University of Msila
M'sila 28000, Algeria
nouri.benaidja@univ-msila.dz

***Abstract*—This paper deals with the method of reference model applied to the identification of parameters of nonlinear systems using three society-inspired heuristics. Heuristics used in this paper are harmonic search HS, teaching-learning algorithm TLA and cultural Algorithm CA. Furthermore, hybridization of CA with artificial immune system AIS is investigated in order to obtain high performance meta-heuristic CA-AIS. The resulting algorithm called CA-AIS is oriented to estimate the parameters of Tanagi-Sugeno fuzzy model of nonlinear systems. The CA-AIS is executed in two different phases. First, the belief space is built considering situational knowledge and domain knowledge only, the event of dramatic drop of objective function is recorded and exploration-exploitation balance is preserved. Second, the historical knowledge is added to the belief space to exploit the information collected at the previous phase and enhance the intensification capability. Finally, the algorithms are tested via parameter identification of the fuzzy model of Narendra-Parthasarathy nonlinear system. Identification results point the superiority of the proposed algorithm over the typical genetic algorithm and genetic algorithm hybridized with simulated annealing GA-SA.**

***Index Terms*—optimization, fuzzy system, Harmony Search, Teaching Learning Algorithm, Cultural Algorithm.**

I. Introduction

MANY industrial systems generally are nonlinear, uncertain and disturbed by noise. So, fuzzy logic may offer advantage over traditional scheme looking back its structural tolerance, where heuristics are applied to adjust the parameters of the fuzzy model [1]. Many existing bio-inspired approaches are developed for optimization of non-linear systems [2], [3]. Recently, emerging class inspired from social phenomena exhibits interesting characteristics are applied to multiple optimization problems. In this paper three society inspired algorithms are proposed for optimization of the Takagi-Sugeno-fuzzy model widely used for non-linear systems. Furthermore, cultural algorithm is hybridized with artificial immune system and compared with a typical evolutionary algorithm.

II. Fuzzy model

A. Fuzzification

Fuzzification brings knowlege to inputs by the concepts of fuzzy sets and linguistic variables. For our zero-order-Takagi-Sugeno-system we choose triangular membership functions with minimal configuration for inputs e and Δy, as shown in Fig. 1a and Fig. 1b. A_{ij} and B_{ij}. Fuzzy linguistic values are defined as :

$$A_j^i = max(min[\frac{e_i - a}{b - a}, \frac{c - e_i}{c - b}], 0) \quad (1)$$

$$B_j^i = max(min[\frac{\Delta y_i - a}{b - a}, \frac{c - \Delta y_i}{c - b}], 0) \quad (2)$$

Fuzzy singleton C_i is defined for output y as in Fig. 1c.

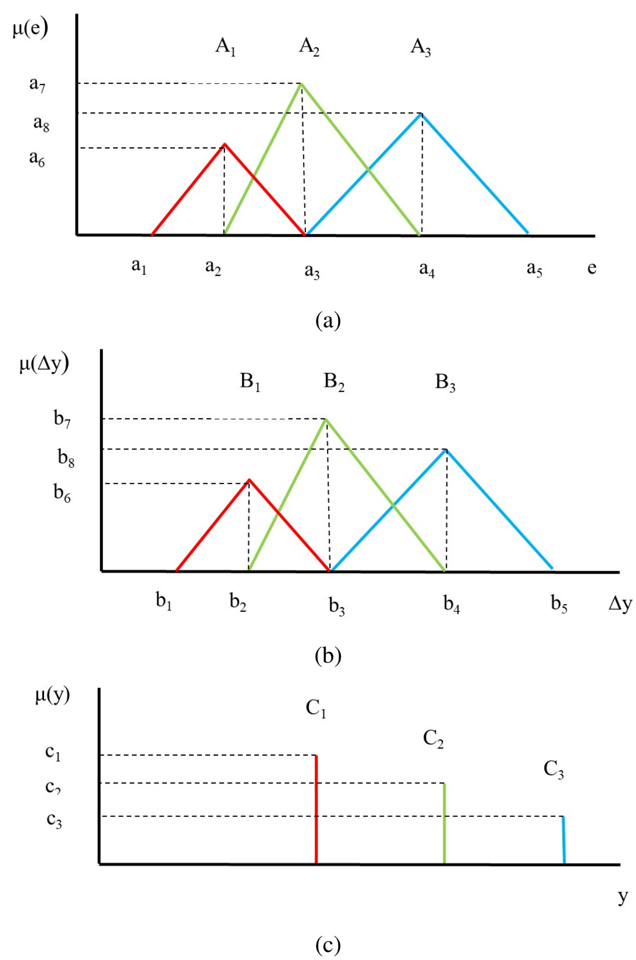

Fig. 1: Parameterized membership functions.

979-8-3315-6387-5/25/$31.00 ©2025 IEEE

B. Rule base

Only three activated rules are retained expressed as follows:

$$\begin{aligned} &If\ e\ is\ A1\ and\ \Delta y\ is\ B1\ then\ y\ is\ C1 \\ &If\ e\ is\ A2\ and\ \Delta y\ is\ B2\ then\ y\ is\ C2 \\ &If\ e\ is\ A3\ and\ \Delta y\ is\ B3\ then\ y\ is\ C3 \end{aligned} \tag{3}$$

C. Inference engine and defuzzification

Given an input dataset $x = (x_1, x_2) = (u, \Delta y)$, for the rule i the firing strength $\beta(x)$ of is expressed as:

$$\beta_{\mathrm{i}}(x) = \prod_{j=1}^{2} \mu_{\mathrm{ij}}(x_{\mathrm{j}}) \tag{4}$$

The weighted average defuzzification method is chosen for the output:

$$y = \frac{\sum \beta_{\mathrm{i}}(x) c_{\mathrm{i}}}{\sum \beta_{\mathrm{i}}(x)} \tag{5}$$

III. HARMONY SEARCH (HS)

Harmony search is a jaz-inspired heuristic optimization algorithm. It is based on the music improvization process and search for a perfect state of harmony [4].

A. Parameters initialization

The *HS* parameters are defined, namely Harmony Memory Size (*HMS*)y; Harmony Memory Considering Rate (*HMCR*); Pitch Adjusting Rate (*PAR*); and Number of Improvisations (*NI*). Also, feasible solutions region is specified.

B. Harmony memory initialization

Harmony memory *HM* matrix is filled with *HMS* vectors of randomly generated numbers presenting the varibles to be optimized :

$$HM = \begin{pmatrix} x_1^1 & . & . & . & x_1^N \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ x_1^{HMS} & . & . & . & x_N^{HMS} \end{pmatrix} \tag{6}$$

The corresponding fitness are computed.

C. Harmony memory improvisation

A new harmony vector, $x' = (x'_1, x'_2, ..., x'_N)$, is built acording to the rules:

- Random Selection: A new harmony vector, $x' = (x'_1, x'_2, ..., x'_N)$, is randomly generated from the feasible search range with a probability of (1- *HMCR*). Random selection is useful for memory initialization.
- Memory Consideration: The pitch i is selected, with a probability of *HMCR*, from the harmonies belonging to *HM*.

$$x'_i = \begin{cases} x'_i \epsilon \{x_i^1, x_i^2 ... x_i^{HMS}\} \ with\ probability\ HMCR \\ x'_i \epsilon X_i \end{cases} \tag{7}$$

D. Pitch Adjustment

The harmony vector, x' selected in the previous step is modified with probability of *PAR*:

$$x'_i = \begin{cases} yes & with\ probability\ PAR \\ no & with\ probability\ (1 - PAR) \end{cases} \tag{8}$$

If the pitch adjustment decision for x' is yes every component of the vector, $x' = (x'_1, x'_2, ..., x'_N)$ is shifted to a neighbour value with respect to the feasible region. The process is achieved by adding certain amount to the value:

$$x'_i = x'_i \pm bw \tag{9}$$

where, '*bw*' is the arbitrary distance bandwidth. The value of (1-*PAR*) sets the rate of the vector remaining at its original value [5].

- Updating HM: If the new harmony vector, $x' = (x'_1, x'_2, ..., x'_N)$, is better than the worst harmony in the *HM*, acording to its fitness, the new harmony is stored in the *HM* and the existing worst harmony is discrded.
- Steps 3 and 4 are repeated until the maximum number of improvisations is reached. Finally, the best harmony memory vector is considered to be the best solution to the problem.

IV. TEACHING-LEARNING ALGORITHM (TLA)

TLA emulates the mechanism of school learning based on passing knowledge within a classroom environment [6]. A teacher is considered as a source of knowledge, students or learners learn from teacher to improve their skills. Students also enhance their knowledge by interacting with each other. Finally, the ability of learner is evaluated by his grades in the class.

A. Teacher Phase

Firstly, a matrix of size [$nl \times ndv$] is built, where nl is the number of learners and ndv presents the subjects studied by the learners analogous to the number of design variables. Let $X_{\mathrm{i}} = (x_{\mathrm{i},1}, x_{\mathrm{i},2}, ..., x_{\mathrm{i,ndv}})$ is the i^{th} learner. The best learner is considered as the teacher $X_{teacher}$ and is taken as current optimal solution. The mean individual of current class X_{mean} is computed and the whole population of learners is updated as given by:

$$X_{new,i} = X_{old,i} + r \times (X_{teacher} - TF \times X_{mean}) \tag{10}$$

where $X_{new,i}$ and $X_{old,i}$ are the new and old positions of the learner, $X_{teacher}$ is the current teacher position, r is a random number in the range [0, 1]. TF is the teaching factor which may be either 1 or 2 and is set according to the equation:

$$TF = round[1 + r] \tag{11}$$

$X_{old,i}$ substitute $X_{new,i}$ only if it is more fitted. otherwise $X_{old,i}$ will keep its value for the next step.

B. Learner Phase

Learners enhance their ability by group discussions and formal communication. Interaction between different learners maintains the diversity of population, avoids premature convergence, and strengthens the global search ability. An i^{th} learner is updated according to the equation given by:

$$X_{new,i} = \begin{cases} X_{old,i} + r(X_{old,i} - X_{old,j}); & f(X_{old,i}) < f(X_{old,j}) \\ X_{old,i} + r(X_{old,j} - X_{old,i}); & otherwise \end{cases} \quad (12)$$

where $X_{old,j}$ are the old individuals, $X_{new,i}$ are the new individuals and f is the objective function. The process of learning terminates when the maximum number of iterations is reached.

V. CULTURAL ALGORITHM (CA)

The Cultural Algorithm uses human knowledge experience to enhance group biliefs. CA evolves through two levels: belief space and population space [7]. The Experience acquired by individuals at the population space according to the acceptance function is used to generate problem solving knowledge injected then into the belief space. This knowledge helps control the evolution of individuals by means of the influence function.

A. Initialization

The initial population is composed of N parent individuals who are randomly created.

$$x_j^i = \begin{pmatrix} a_1 & a_2 & . & . & . & a_8 \\ b_1 & b_2 & . & . & . & b_8 \\ c_1 & & & c_2 & & c_3 \end{pmatrix} \quad (13)$$

Each individual should satisfy the constraints:

$$l < x < u \quad (14)$$

Individuals are sorted according to their fitness. Thus, for our fuzzy system it is defined to be the root-mean-squared error (RMSE), i.e.:

$$f = \sqrt{\frac{\sum_{k=1}^{N} \Delta y}{n}} \quad (15)$$

Where n is the number of computed values.

B. Updating the Belief Space – acceptance function

The Belief Space is organized in three knowledge domains as:

- Situational domain: The situational knowledge S consists of a current best exemplar individual and the elite S is updated as:

$$S = \begin{cases} x_{best}; & f(x_{best} \leq f(S) \\ S; & otherwise \end{cases} \quad (16)$$

  The selected individuals by the acceptance function are used to determine the feasible range useful for the normative knowledge.

- Normative domain: Firstly minimum x_i and maximum x_k values for parameter j among accepted individuals in the current generation are selected. Then the feasible range of normative knowledge is updated as follows,

$$\begin{aligned} l_j^{t+1} &= \begin{cases} x_i^t; & x_i^j \leq l_j^t \ or \ f(x_i) < L_j^t \\ l_j^t; & otherwise \end{cases} \\ L_j^{t+1} &= \begin{cases} f(x_i); & x_i^j \leq l_j^t \ or \ f(x_i) < L_j^t \\ L_j^t; & otherwise \end{cases} \\ u_j^{t+1} &= \begin{cases} x_i^t; & x_i^j \geq u_j^t \ or \ f(x_i) < U_j^t \\ u_j^t; & otherwise \end{cases} \\ U_j^{t+1} &= \begin{cases} f(x_i); & x_i^j \geq u_j^t \ or \ f(x_i) < U_j^t \\ U_j^t; & otherwise \end{cases} \end{aligned} \quad (17)$$

  Where, l_j^t is the lower bound for parameter j at generation t and L_j^t is its finess. u_j^t is the upper bound for parameter j at generation t and U_j^t its fitness.

- Historical domain: The algorithm is running in two stages : in the first stage the event of dramatic drop in fitness function is recorded and meadwhile the good candidate will be recovered in the second stage. This procedure is anti-analogous to the concept of tabu search.

C. Influence function

The influence function manipulates the knowledge stored in the belief space aiming the generation of new individuals. The individual is slightly perturbed if that individual is in a promising region:

$$x_{i,j}^{t+1} = \begin{cases} x_{i,j}^t + |(u_j - l_j) * N(0,1)|; & if \ x_{i,j}^t < l_j^t \\ x_{i,j}^t - |(u_j - l_j) * N(0,1)|; & if \ x_{i,j}^t > u_j^t \\ x_{i,j}^t + \rho * (u_j - l_j) * N(0,1); & otherwise \end{cases} \quad (18)$$

Where $N(0,1)$ represents uniformly distributed Gaussian random number and ρ decreasing variable.

If an individual is in an unpromising region, it should be pushed to a more promising region. In this case, the situational knowledge applies.

$$x_{i,j}^{t+1} = \begin{cases} x_{i,j}^t + |(u_j - l_j) * N(0,1)|; & if \ x_{i,j}^t < S_j^t \\ x_{i,j}^t - |(u_j - l_j) * N(0,1)|; & if \ x_{i,j}^t > S_j^t \\ x_{i,j}^t + \rho * (u_j - l_j) * N(0,1); & otherwise \end{cases} \quad (19)$$

VI. ARTIFICIAL IMMUNE SYSTEM (AIS)

The biological immune system is an intelligent recignition and protection process by having the ability of distiguish between intruders such as bacteria, viruses, and other parasites (called antigens) and the substances that constitute self [8].

The main steps of the artificial immune system are described as:

1: Initialization: Memory set is composed from the whole antibody population P. It is intially randomly generated and uniformly distributed over the search space.

2: **Repeat.**

3: Affinity evolution: affinity is the match capability between antigen and antibody represened by fitness function.

4: Clonal selection: C_s best antibodies are selected, according to their affinity.

5: Replication (clone): C_s best antibodies of the population leading to a temporary population of clones (C).

6: Hypermutation scheme: A new population of antibodies (C') is generated by the hypermutation process. Hypermutation is proportional to the affinity given by:

$$C' = C + \sigma^{-1} * e^{-f} * N(0,1) \tag{20}$$

where $N(0,1)$ is a Gaussian random variable of zero mean and unitary variance, σ is a parameter that controls the decay of the inverse exponential function, and f is the normalized affinity of the C.

7: Re-selection: Slected fittest members of P are replaced by other improved members of C' and the new antibody population is generated (P_r).

8: Replace dc worst antibodies in (P_r) by randomly generated antibodies.

9: cycle = cycle + 1.

10: **Until** a termination is satisfied.

VII. HYBRID CULTURAL ALGORITHM (CA-AIS)

Cultural algorithm is hybridized with artificial immune system CA-AIS. Influence function is simulated to clonal selection. Thus, the clone size (N_c) is set equal to the size of the belief space. Hypermutation scheme is added after the influence function. The main steps of the CA-AIS are summarized as:

1: t = 0.
2: Initialize population $P(t)$.
3: Initialize belief space $B(t)$.
4: **Repeat.**
5: Evaluate $P(t)$.
6: Update($B(t)$), accept($P(t)$).
7: Generate ($P(t)$), influence$B(t)$).
8: Hypermutation scheme.
9: Select $P(t)$ from $P(t-1)$.
10: $t = t + 1$.
11: **Until** a termination is satisfied.

VIII. SIMULATION RESULTS

Optimization algorithms are tested considering *Narendra-Parthasarathy* nonlinear system [**?**]. The Narendra-Parthasarathy system is described as follows:

$$y(k+1) = \frac{y(k)}{1+y^2(k)} * + u^2(k) \tag{21}$$

where $u(k)$ and $y(k)$ are the input and output of the system, respectively, at the iteration k. Performances of the algorithms are then evaluated on a test set of 250 points generated from a sinusoidal input signal of the form:

$$u(k) = sin(\frac{\pi k}{50}) * cos(\frac{\pi k}{30}) \tag{22}$$

The fitness function f for performance evaluation is defined to be the root-mean-squared error (RMSE), i.e.:

$$f = \sqrt{\frac{\sum_{k=0}^{249}(y_m(k+1) - y(k+1))^2}{250}} \tag{23}$$

TABLE I: PARAMETER SETTING OF OPTIMIZATION ALGORITHMS

Algorithm	**Parameters**
GA	pcross = 0.9, pmut = 0.3, popsize = 40.
GA-SA	pcross = 0.9, pmut = 0.3, α = 0.9, σ = 0.01, popsize = 40.
HS	HMCR = 0.85, PARmin = 0.2, PARmax = 2, BWmin = 0.45, BWmax = 0.9, popsize = 20.
TLA	popsize = 20.
CA	λ max = 1, popsize = 40, beliefsize = 10.
CA-IS	λ max = 1, σ = 100, popsize = 40, beliefsize = 10.

TABLE II: OPTIMIZED PARAMETER MATRIX FOR FUZZY RULE BASE

Algorithm	**Parameters: a1, a2, a3, a4, a5, a6, a7, a8, b1, b2, b3, b4, b5, b6, b7, b8, c1, c2, c3.**
GA	-4.526, -4.524, -4.518, 3.398, 9.112, -20.818, -20.805, -20.802, 18.138, 41.113, 1.759, 4.903, 1.117, 19.625, 4.058, 1.167, 40.099, -6.147, 18.764.
GA-SA	-1.840, -1.579, 0.728, 1.219, 1.744, -5.8, -5.737, 1.456, 1.496, 2.288, 1.945, 1.325, 1.217, 1.416, 1.242, 2.576, -2.775, 0.909, 2.775.
HS	-1.554, -0.907, 0.976, 1.142, 1.423, -37.652, -2.886, 36.974, 36.989, 39.379, 0.318, 0.608, 1.536, 2.155, 2.212, 1.625, -1.277, 1.252, 7.008.
TLA	0.016, 0.065, 0.081, 0.082, 0.126, -1.417, -1.085, -0.478, 0.961, 1.809, 0.284, 0.456, 0.06, 3.28, 0.001, 1.628, -0.441, -0.596, -0.0005.
CA	-3.373, -2.521, -0.808, 1.965, 2.622, -11.81, -7.435, -1.18, 13.346, 25.757, 0.23, 1.001, -0.375, 5.697, -1.781, 4.404, -26.689, -0.858, 12.682 .
CA-IS	-2.381, -0.737, 0.822, 1.364, 1.464, -11.971, -2.626, 8.381, 13.762, 17.696, -0.661, 1.474, -0.126, 5.111, -1.611, 5.225, -1.297, 1.06, 19.282

TABLE III: PARAMETER SETTING OF OPTIMIZATION ALGORITHMS

Algorithm	**Best fitness**
GA	0.182936
GA-SA	0.174393
HS	0.183018
TLA	0.473035
CA	0.238671
CA-IS	0.210504

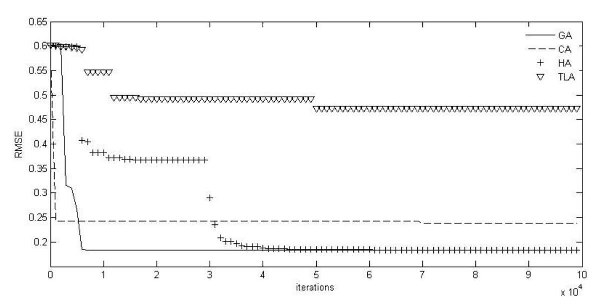

Fig. 2: Strength of parameter identification of Narendra-Parthasarathy system.

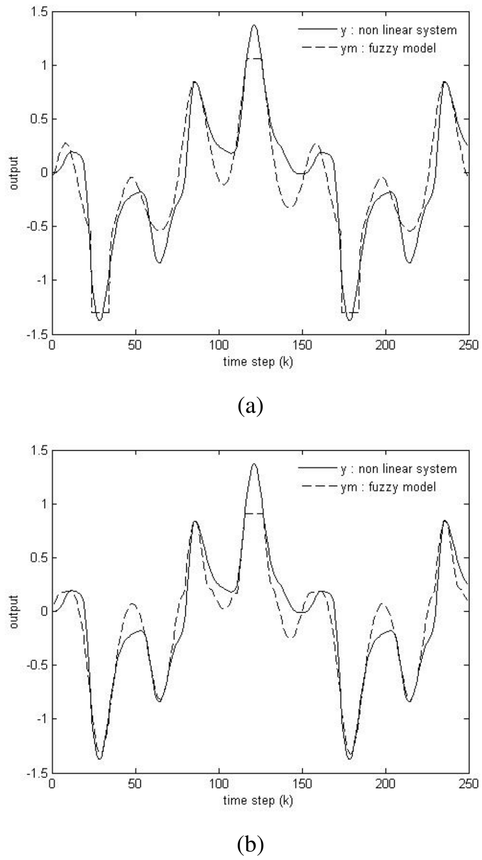

Fig. 3: Outputs resulting from identification by hybrid (a) CA-AIS and (b) GA-SA.

CA-AIS has been tested using several combinations: populations of different sizes (from 5 to 50), hypermutation rate from 0.01 to 100. Two ratios for the size of belief space are tested: quarter and half of the population size. Poor results are observed with populations of small size under 10, above this value the fitness of individuals improve significantly and reach the best value for *popsize* of 40. Above this value the performances of the algorithm decrease. No noticeable improvement of the value of fitness function is observed when the size of belief space is increased from quarter to the half of population size at the expense of execution time. The same remark is observed for hypermutation rate σ. The values of parameters of the algorithms giving the best resulting fitness are reported in table I. Where *pcross* is the crossover probability, *pmut* is the mutation probability, *popsize* is the population size of and beliefsize the size of belief space.

The parameters of the fuzzy system resulting from different algorithms are given in table II. The best fitness after convergence are given table III. When the convergence is reached the fitness values given by hybrid CA-AIS and HS are 0.21 and 0.183 respectively. So, generally society-inspired algorithms are efficient for parameter identification of a nonlinear system. The main advantage of teaching learning algorithm is that is parameters free, but its intensification-diversification mechanism is not sufficiently efficient to compete with the other algorithms. Nevertheless, the hybridization of TLA with simulated annealing did no longer improve the value of the fitness function. The convergence of the fitness function for the algorithms is depicted in Fig. 2.

IX. CONCLUSION

In this paper, an effective algorithm based on the hybridization of biological immune system and cultural algorithm has been developed for the purpose of determining the optimal parameters of the fuzzy model of nonlinear systems. Numerical simulations demonstrate that the proposed algorithms possess the merits of global exploration and accurate intensification in solving the parameter identification problem. For further works, we want to apply CA-AIS to solve more practicable engineering optimization problems and find some other efficient mechanisms to improve the TLA.

REFERENCES

[1] L.A. Zadeh, "Fuzzy sets," Inf. Control, vol. 8, pp. 338–353, 1965.

[2] N. Benaïdja, N. Khenfer, "Identification of Asynchronous Machine Parameters by Evolutionary Techniques," Elec. Pow. Comp and Syst, vol. 34, no. 12, pp. 1359–1376, 2006.

[3] N. Benaïdja, "Soft computing identification techniques of asynchronous machine parameters: evolutionary Strategy and chemotaxis algorithm," Turk. J. Elec. Eng and Comp. Sci, vol. 17, no. 1, pp. 69–85, 2009.

[4] Z.W. Geem, J.H. Kim and G.V. Loganathan, "A new heuristic optimization algorithm: Harmony search," Simulation, vol. 76, no. 2, pp. 60-68, 2001.

[5] A. Turky, S. Abdullah and A. Dawod, "A dual-population multi operators harmony search algorithm for dynamic optimization problems," Comput. Ind. Eng, vol. 117, pp. 19–28, 2018.

[6] R. Rao, V. Savsani, and D. Vakharia, "Teaching-learning based optimization: an optimization method for continuous non-linear large-scale problems," Inf. Sci, vol. 183, pp. 1-15, 2012.

[7] R-G. Reynolds, Cultural Algorithms: Tools to Model Complex Dynamic Social Systems, Wiley-IEEE Press, 2020.

[8] J.P. Tien, T.H.S. Li, "Hybrid Taguchi-chaos of multilevel immune and the artificial bee colony algorithm for parameter identification of chaotic systems," Comput. and Math. with Appl, vol. 64, pp. 1108–1119, 2012.

[9] K.S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," IEEE Trans. Neural Netw, vol. 1, no. 1, pp. 4–27, 1990.