Optimization of Interval Type-2 Fuzzy Logic Controller Using Real-Coded Quantum Clonal Selection Algorithm

Esra Satir, Ekrem Baser*
Department of Computer Engineering, Duzce University,
81620, Duzce, Turkey
ekrembaser@duzce.edu.tr

***Abstract*—In recent years, quantum computing has gained immense popularity with the production of real quantum computers. Researchers have developed quantum-inspired evolutionary algorithms (QIEAs) to solve combinatorial optimization problems and have obtained successful results. As a special case of QIEAs, real-coded quantum evolutionary algorithm (RCQEA) is used in the optimization of high-dimensional complex problems. In this study, a novel mechanism of the quantum rotation gate (QRG) that is used to determine the rotation angle of the qubit in the RCQEA is introduced and implemented to accelerate the evolutionary process and increase the possibility of finding the optimal solution. Moreover, the skeleton of RCQEA is modified by using a clonal selection mechanism, and the real-coded quantum clonal selection algorithm (RCQCSA) is developed. Our proposed QRG accelerates the convergence speed of the algorithm. The main purpose of this study is to present a more effective algorithm that inspires quantum computing principles for optimizing the interval type-2 fuzzy logic controller (IT2FLC) membership functions (MFs). In this study, four different comparisons are made with these two different algorithms that have the original version of QRG and our proposed QRG. Optimized IT2FLC provides stabilization of the inverted pendulum system. The results show that the RCQCSA having our proposed QRG outperforms RCQEA in stabilizing the inverted pendulum system by optimizing the IT2FLC parameters.**

***Index Terms*—Clonal selection mechanism; Interval type-2 fuzzy logic; Optimization; Quantum evolutionary algorithms.**

I. INTRODUCTION

Most traditional decision-making systems are based on bivalent logic, which is the basis of the logic concept. In the bivalent logic approach, known as classical logic, an entity is certainly either a member of the set or not. It means that all of the possible denotations of propositions are categorized as true or false as mathematically 1 or 0. In 1965, Professor Lotfi A. Zadeh introduced the concept of fuzzy logic [1]. Until 1975, Zadeh broadened this theory by establishing fuzzy similarity relations, linguistic hedges, and fuzzy decision-making [2]–[4]. Fuzzy logic starts with the concept of a fuzzy set (FS) that contains elements with only a partial degree of membership. The main difference of the fuzzy logic theory from the classical theory is that an entity can partially belong to a fuzzy set and can partially belong to another fuzzy set. Therefore, a FS can be interpreted as a combination of some distinct elements with a varying degree of membership.

FS introduced by Zadeh is known as type-1 FS. Also, a system that has at least one type-1 FS, is called “type-1 FLC”. Type-1 FLC employs ordinary sets, which represent uncertainty with crisp values in the range of [0, 1]. Once something is uncertain, the exact value of the entity cannot be known and cannot be assessed as a crisp value [3], [4]. Therefore, another type of membership function (MF), called a “type-2 MF”, and in this case another type of FS, called a “type-2 FS” arose from the necessity of dealing with these uncertainties as an extension of a type-1 FS [5]. A type-2 FS has membership degrees that are themselves fuzzy. Namely, the grade of a type-2 MF can be any subset of [0, 1]. The system that has at least one type-2 FS is called “type-2 FLC”.

A type-2 FLC is more capable of dealing with linguistic uncertainties by modelling vagueness and unreliability of information [6], [7]. Thanks to its quite impressive features, it has been used in many areas, such as pattern recognition [8], control [9], classification [10], edge detection [11], etc. A special case of type-2 FLCs, interval type-2 fuzzy logic controller (IT2FLC) have attracted much research interest, especially in control applications. It has been shown in many studies that IT2FLC achieves better control performance due to the additional degree of freedom provided by its special MFs [12], [13]. It is explained in [14], [15] that IT2FLC is generally more robust than its type-1 counterparts.

One of the IT2FLC components that is briefly explained in the next section is the knowledge base (KB). KB consists of FSs that are input and output MFs and fuzzy rules that map input MFs to output MFs. Fuzzy rules and MFs are either provided by human experts or learned from sample data. The system to be controlled with IT2FLC can be highly complex and has non-linear system dynamics. Without such an expert KB, the system cannot be controlled effectively. In this case, fuzzy rules or MFs need to be optimized to control the system as desired.

Tuning the parameters of IT2FLC and obtaining an expert KB for the controlled system is a popular optimization problem due to IT2FLC’s high capability on dealing with

Manuscript received 21 November, 2021; accepted 16 April, 2022.

non-convex or non-differentiable objective functions.

Artificial immune systems (AIS) are distributed and computational systems inspired by the principles of natural systems and also the human immune system [16], [17]. In terms of running mechanism, there exist different types of metaheuristic algorithms based on AIS. One of them, the Clonal Selection Algorithm (CLONALG), is inspired by the principle of clonal selection of the immune system [18]–[20]. Generally, this algorithm models the basic features of an adaptive immune response to an antigenic stimulus. It explains how an immune response mounts when an antigenic pattern is recognized by the B-cells. It consists of three main processes: selection, proliferation, and differentiation. The proliferation process is succeeded by cloning B-cells. The number of clones of B-cells is directly proportional to their affinity degree for the antigen. The clones are subjected to the hypermutation operator inversely proportional to their affinity. A detailed pseudo-code and the CLONALG flowchart can be found in [20].

In the early 2000s, the quantum-inspired evolutionary algorithm (QIEA) was presented as an optimization algorithm that takes inspiration from quantum computing to evolve a probability distribution. The algorithm uses a string of quantum bits (qubits) that hold the sampling probability of a one or a zero [21], [22]. Therefore, it was applied on only binary combinatorial optimization problems, and then modified versions of the algorithm were applied on real-valued problems. To deal with complex functions with high dimension, the real-coded quantum evolutionary algorithm (RCQEA) was proposed in [23], [24]. RCQEA is based on concepts and principles of quantum computing, such as qubits and superposition of states by using triploid chromosomes whose alleles are composed of a real variable and a pair of probability amplitudes of the corresponding qubit state. The algorithm uses the complementary double mutation operator (CDMO), quantum rotation gate (QRG), and discrete crossover (DC) for diversity of solution and the Hill-climbing selection (HCS) for accelerating the convergence speed. RCQEA has brought a new perspective on optimization problems by inspiring quantum computing principles.

This study focuses on the implementation of a more effective quantum-inspired algorithm than RCQEA in tuning the parameters of IT2FLC. We propose the Real-Coded Quantum Clonal Selection Algorithm (RCQCSA) for the optimization of IT2FLC parameters. We also improved the QRG of the RCQEA so that the algorithm finds the expected solutions in the solution space more consistently. We use an IT2FLC controlled inverted pendulum system that is a highly non-linear unstable system to test the performance of our proposed algorithm and RCQEA.

The organization of this paper is given as follows. Section II introduces the related works. Section III briefly presents the structure of the IT2FLC. In Section IV, the problem statement and the mathematical model of the IT2FLC controlled inverted pendulum system are presented. Section V presents the RCQEA algorithm briefly. The proposed RCQCSA algorithm for IT2FLC optimization is explained in Section VI. In Section VII, some simulation results are demonstrated and discussed comparatively. Conclusions are drawn in the final section.

II. RELATED WORKS

Numerous methods such as the genetic algorithm [25], [26], neural networks [27], particle swarm optimization (PSO) [28], [29] harmony search algorithm [30], imperialist competitive algorithm [31], RCQEA [32]–[36], firefly algorithm and galactic swarm optimization [37], bee colony algorithm [38], slime mould algorithm [39], and shark smell optimization [40] have been used in IT2FLC optimization to improve its behaviour on the system.

As an optimization algorithm, CLONALG has been used to solve different types of optimization problems [41]–[43]. There are many studies for FLC optimization with CLONALG, such as in [44], [45].

There are many studies on RCQEA to improve it [46]–[48], or create hybrid evolutionary algorithms [49]–[52]. The authors in [41] changed the qubit rotation magnitude of the algorithm with a constant magnitude. The authors in [53] replaced the quantum rotation gate by a like-approximation operator. In [54], a better way to determine rotation angle has been proposed, in addition to using real qubits instead of binary qubits. The authors in [55] proposed PSO based quantum evolutionary algorithm by changing the strategy of updating the rotation angle. The authors in [56] used a different method for the magnitude of the QRG rotation angle. The RCQEA implementation method to optimize the FLC in the literature and shortcomings [32]–[36] is presented below.

In [32]–[36], the whole domain of inputs and outputs in IT2FLC is divided into equal-sized intervals and MFs are generated in these intervals. The rules of IT2FLC should also be optimized for the system, which is controlled by IT2FLC designed with MFs generated at these intervals, to reach equilibrium quickly. Because, if the inputs of the system are not in the range, where MFs representing the linguistic terms in IT2FLCs rules that bring the system to equilibrium the fastest are generated, these rules cannot be processed, and the system cannot reach equilibrium quickly.

In [32]–[36], RCQEA was applied on the genes that are formed by using IT2FLC rules. MFs are generated and encoded in chromosomes according to the determined intervals of MFs mentioned in the antecedents and consequents of the rules in IT2FLC. The cost of a chromosome is evaluated by designing the MFs of the IT2FLC by using the values that stored in the genes of the corresponding chromosome and getting the response of controlled system. However, this strategy is not effective. Because if the MFs, generated according to the linguistic terms mentioned in the antecedents and consequents of the rules, belong to the same linguistic terms, the MFs generated later will overwhelm the first generated MF while evaluating the cost of the chromosome. Namely, IT2FLC is designed by using the post-generated MFs belonging to the same linguistic terms. Therefore, although the algorithm generates a more useful MF for IT2FLC with crossover and mutation operators than a MF belonging to the same linguistic term, this useful MF is ignored during the evaluation process of the chromosome if it is not the last generated MF belonging to the same linguistic term.

In [32]–[36], gaussian MFs are used in IT2FLC to be optimized. The point of the Gaussian MF having the maximum degree of membership is always the midpoint of the MF. Since MFs in the system can be generated between the intervals determined for them, Gaussian-type MFs that will bring the system to equilibrium in the fastest way, may not be generated in all cases.

In summary, the main contributions of this study are as follows:

1. The QRG of the RCQEA algorithm is improved by modifying the rotation angle determination strategy. The rotation angle determined in the original version of QRG [23], [24] can be aimless by reaching values of large magnitudes in some cases of alpha and beta values of the allele, while it should be maximum $\pi/2$ as detailed in [47], [48]. We propose a different rotation angle determination strategy to solve this problem and accelerate the convergence speed of the algorithm. We also compare the effect of our QRG and the QRG proposed in [47] on the algorithm.
2. In this study, one MF is generated for each linguistic term in the IT2FLC in the ranges where they could be generated and encoded into chromosomes. Therefore, the overriding problem is solved.
3. In this study, we propose the RCQCSA, which we created by improving the mutation and quantum rotation strategy used in the RCQEA algorithm for the optimization of IT2FLC MFs and basing the algorithm on the principles of clonal selection, as a contribution to the literature. Although the same initial population is used for RCQEA and RCQCSA in IT2FLC optimization and there is no crossover operation in RCQCSA, RCQCSA performs better and converges faster than RCQEA.
4. In this study, skewed Gaussian MFs are generated in their corresponding regions. Due to this type of MFs, each input in the range in which it is generated can be defined by a skewed Gaussian MF with a maximum membership degree.
5. In this study, the regions where IT2FLC MFs are generated are determined by a different strategy. Therefore, only the MFs of IT2FLC are optimized without the rules of IT2FLC and the system can reach equilibrium quickly.

III. A BRIEF STRUCTURE OF IT2FLC

IT2FLC contains at least one interval type-2 fuzzy set (IT2FS), which is a special case of type-2 FSs. IT2FLC reduces the computational complexity of the general type-2 FLC [3], [57]–[60]. An IT2FS $\tilde{X}$ is characterized by its MF $\mu_{\tilde{X}}(x,u)$ and as expressed in (1), where x is called as the "primary variable" and it has domain $D_{\tilde{X}}$ for IT2FSs

$$\tilde{X} = \int_{x\in D_{\tilde{X}}} \int_{u\in J_x \subseteq [0,1]} \mu_{\tilde{X}}(x,u)/(x,u). \tag{1}$$

The double integral sign denotes the union over all admissible x and u. J_x denotes the primary membership of x. An IT2FS is bounded from above and below by two type-1 FSs, namely, upper MF (UMF) and lower MF (LMF). A Gaussian-type-2 MF is shown in Fig. 1. The bounded region between UMF and LMF is called "footprint of uncertainty" (FOU) [57].

An IT2FLC consists of five components, called "fuzzifier", "interference engine", "KB" which contain the rule base and database, "type reducer", and "defuzzifier" as shown in Fig. 2 [61]. The rule base consists of N fuzzy rules expressed as IF-THEN linguistic statements. Fuzzy rules determine the relationships between the inputs and outputs of the IT2FLC. For the system having k inputs and l outputs, fuzzy rules can be created as in (2)

$$\begin{aligned} \tilde{R}^n &: IF\ x_1\ is\ \tilde{X}_1^n\ ...\ and\ x_k\ is\ \tilde{X}_k^n,\ then \\ &y_1\ is\ \tilde{Y}_1^n\ ...\ and\ y_l\ is\ \tilde{Y}_l^n\ \ n = 1, ..., N. \end{aligned} \tag{2}$$

Fuzzifier block computes the membership interval of each $x_i^{'}$ on each $\tilde{X}_1^n$ as $\left[\underline{\mu}_{\tilde{X}_i^n}(x_i^{'}), \bar{\mu}_{\tilde{X}_i^n}(x_i^{'})\right]$, $i = 1, 2, ..., k$, $n = 1, 2, ..., N$. Inference block calculates the rule firing interval of the n^{th} rule as in (3) by using (4) and (5):

$$F^n(x') \equiv \left[\underline{f}^n, \bar{f}^n\right], n = 1, 2, ..., N, \tag{3}$$

$$\underline{f}^n = \left[\underline{\mu}_{\tilde{X}_1^n}(x_1^{'}) * ... * \underline{\mu}_{\tilde{X}_k^n}(x_k^{'})\right], \tag{4}$$

$$\bar{f}^n = \left[\bar{\mu}_{\tilde{X}_1^n}(x_1^{'}) * ... * \bar{\mu}_{\tilde{X}_k^n}(x_k^{'})\right]. \tag{5}$$

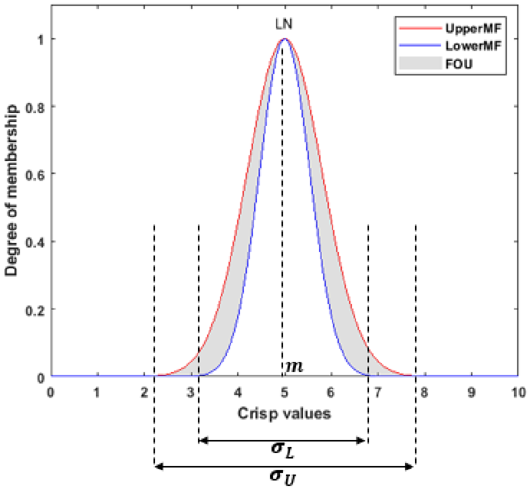

Fig. 1. An interval Gaussian-type-2 MF.

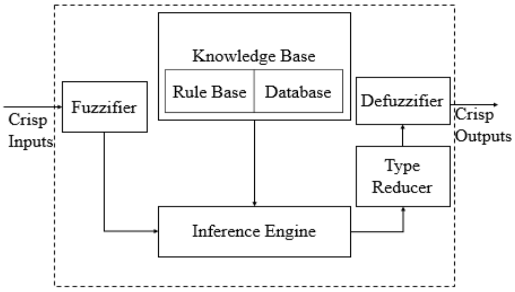

Fig. 2. Schematic diagram of IT2FLC.

The type reducer (TR) block is unique for type-2 FLC and converts IT2FLC to type-1 FLC. There are various TR methods in the literature. The most commonly used TR method is center of sets defined as in (6), where y_l and y_r are defined in (7) and (8):

$$Y_{cos}(x') = \bigcup_{\substack{f^n \in F^n(x') \\ y^n \in Y^n}} \frac{\sum_{n=1}^{N} y^n f^n}{\sum_{n=1}^{N} f^n} = [y_l, y_r], \quad (6)$$

$$y_l = \frac{\sum_{n=1}^{L} \underline{y}^n \overline{f}^n + \sum_{n=L+1}^{N} \underline{y}^n \underline{f}^n}{\sum_{n=1}^{L} \overline{f}^n + \sum_{n=L+1}^{N} \underline{f}^n}, \quad (7)$$

$$y_r = \frac{\sum_{n=1}^{R} \overline{y}^n \underline{f}^n + \sum_{n=R+1}^{N} \overline{y}^n \overline{f}^n}{\sum_{n=1}^{R} \underline{f}^n + \sum_{n=R+1}^{N} \overline{f}^n}. \quad (8)$$

The defuzzification block computes crisp output values of IT2FLC as in (9)

$$y = \frac{y_l + y_r}{2}. \quad (9)$$

IV. APPLIED WORK

The inverted pendulum system has been a popular test problem for classical and contemporary control techniques because of its being open loop unstable system with high non-linear system dynamics. Figure 3 shows the schematic diagram of the system which is formed from a cart, a pole, and a rail to define the position of the cart. The pole is hinged to the centre of the top surface of the cart and can rotate around the pivot in the vertical plane. The cart can move horizontally to the right or to the left on the rail. The system is unstable without control, i.e., the pendulum will fall over if the cart is not moved to balance it. Therefore, the system provides an opportunity to analyse the performance of control techniques for stabilizing the pole vertically. The architecture of IT2FLC optimization and IT2FLC controlled inverted pendulum system is shown in Fig. 4. Given that no friction exists, the dynamics of the inverted pendulum system are modelled by the non-linear differential equations as in (10) and (11):

$$\ddot{\theta} = \frac{(m_c + m_p) g sin\theta - \{F + m_p l w^2 sin\theta\} cos\theta}{\{4/3(m_c + m_p) - m_p (cos\theta)^2\} l}, \quad (10)$$

$$\ddot{x} = \frac{4/3\{F + m_p l w^2 sin\theta\} - m_p g sin\theta cos\theta}{\{4/3(m_c + m_p) - m_p (cos\theta)^2\}}. \quad (11)$$

The variables, which are θ, w, x, v, denote the angle of the pendulum, its angular velocity, the position of the cart and its velocity, respectively. Parameters of the inverted pendulum system are assumed for the simulation as in Table I. They are not attached to a real or physical system.

Basically, there are three types of structure of fuzzy logic controllers: fuzzy proportional derivative (PD) controller, fuzzy proportional integral (PI) controller, and fuzzy proportional integral derivative (PID) controller. The fuzzy PD controller generates the control signal from the error and changes the error rate. Fuzzy PI controller generates incremental control action from error and change rate in error. The fuzzy PID controller generates incremental control action from error, change in error, and acceleration of error. The fuzzy PID controller has three inputs, which will greatly expand the rule base and make the design more complicated. Although the fuzzy PI controller is more practical than the fuzzy PD controller, it has poor performance in system transient response. The fuzzy PD controller is simple in structure and easy to implement [62]. In [63], it is seen that the fuzzy PD controller performs better than the other types of controllers for an inverted pendulum system. Therefore, we use the fuzzy PD controller in this study.

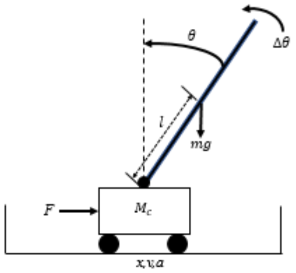

Fig. 3. Schematic diagram of inverted pendulum system.

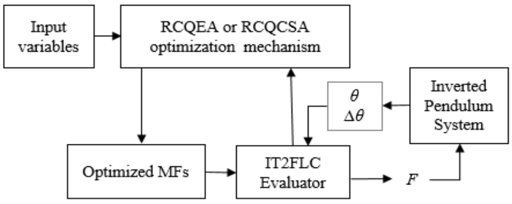

Fig. 4. The architecture of the model.

TABLE I. PARAMETERS OF THE INVERTED PENDULUM SYSTEM.

Symbol	Value	Unit	Parameter Name
m_c	1.0	*kg*	Mass of cart
m_p	0.1	*kg*	Mass of pole
l	0.5	*m*	Length to pendulum center of mass
F	-	*N*	Force applied to cart
θ	0.52	*rad*	Pendulum angle from vertical.
g	9.8	m/s^2	Gravitational acceleration

V. RCQEA ALGORITHM

Based on the concepts and principles of quantum computing, QIEA has been a very strong competitor to classical evolutionary algorithms in optimization problems. QIEAs are more suitable for combinatorial optimization problems than numerical optimization problems [64], [65].

RCQEA is proposed to solve global numerical optimization problems with continuous variables in [23], [24]. The main difference of the RCQEA from the QIEA is the construction of the chromosomes. The real-coded triploid chromosomes represented in (12) are used to keep the diversity of the solution

$$q = \begin{bmatrix} x_1 & x_2 & \dots & x_n \\ \alpha_1 & \alpha_2 & \dots & \alpha_n \\ \beta_1 & \beta_2 & \dots & \beta_n \end{bmatrix}. \quad (12)$$

Alleles of the chromosomes are composed of one component x_i of variable vector X and probability amplitudes $(\alpha_i, \beta_i)^T$ of one qubit. The probability amplitudes satisfy the normalization condition defined in the (13), i = 1, 2, …, n where n is the length of the triploid chromosome

$$\alpha^2 + \beta^2 = 1. \quad (13)$$

Here, $|a_i|^2$ and $|\beta_i|^2$ denote the probability that the qubit will be measured in the state $|0\rangle$ and $|1\rangle$, respectively. Therefore, the total probability of the system being observed in either state $|0\rangle$ or $|1\rangle$ is 1.

RCQEA uses Discrete Crossover (DC) repeatedly after a fixed number of generations. All the corresponding alleles of the randomly selected two chromosomes, namely parent chromosomes, are exchanged with 0.5 probability and two new individuals are formed. This operation ensures the diversity of the population. RCQEA uses CDMO that affects only one gene of the chromosome. The operator updates the randomly selected gene using Gaussian mutation as expressed in (14) where $x_{i,max}^t$ and $x_{i,min}^t$ represent upper and lower bounds of the corresponding allele, respectively

$$x_i^{t+1} = x_i^t + (x_{i,\,max}^t - x_{i,\,min}^t) N\left(0, (\sigma_j^t)^2\right). \tag{14}$$

The $N\left(0, (x_j^t)^2\right)$ denotes the Gaussian distribution of mean 0 and variance $(\sigma_j^t)^2$. This value is determined either $\left|\alpha_i^t\right|^2$ or $\left|\beta_i^t\right|^2/5$ as stated in (15)

$$(\sigma_j^t)^2 = \begin{cases} \left|\alpha_i^t\right|^2, \, r = 0, \\ \dfrac{\left|\beta_i^t\right|^2}{5}, \, r = 1. \end{cases} \tag{15}$$

If the new generated value exceeds the limits of the corresponding allele, it is limited to the upper and lower bounds of the corresponding allele using (16)

$$x_i^{t+1} = \begin{cases} 2x_{i,max} - x_i^{t+1}, \, x_i^{t+1} > x_{i,max}, \\ 2x_{i,min} - x_i^{t+1}, \, x_i^{t+1} < x_{i,min}. \end{cases} \tag{16}$$

If the new generated solution is better than the old solution, the probability amplitudes are fixed; otherwise the probability amplitudes are updated by QRG as expressed in (17) and the rotation angle of QRG is determined as in (18):

$$\begin{pmatrix} \alpha_i^{t+1} \\ \beta_i^{t+1} \end{pmatrix} = \begin{pmatrix} \cos(\Delta\theta_i^t) & -\sin(\Delta\theta_i^t) \\ \sin(\Delta\theta_i^t) & \cos(\Delta\theta_i^t) \end{pmatrix} \begin{pmatrix} \alpha_i^t \\ \beta_i^t \end{pmatrix}, \tag{17}$$

$$\Delta\theta_i^t = sgn(\alpha_i^t \beta_i^t)\theta_0 exp\left(\frac{\beta_i^t}{\alpha_i^t + \gamma}\right). \tag{18}$$

VI. METHODOLOGY

In this section, we introduce the basic principle of RCQCSA, which we adapted from RCQEA [23], [24] and CLONALG [20], and the IT2FLC design process using RCQCSA. RCQCSA is based on the clonal selection principles and is inspired by quantum computing techniques. Algorithm 1 shows the pseudo-code of the RCQCSA.

Algorithm 1. Real-Coded Quantum Clonal Selection Algorithm.

```
Input: antibodies, nGenerations, nSelection, nReplaces, probAmplitudes
Output: bestAffinity, antibodies
begin
    affinities ← calcAffinities(antibodies);
    for iter = 1 to nGenerations do
        selectedAntibodies ← select(antibodies, affinities, nSelection);
        clones ← clone(selectedAntibodies, affinities);
        clones ← CDMO(clones, affinities);
        probAmplitudes ← QRG(clones, affinities);
        cloneAffinities ← calcAffinities(clones);
        update(antibodies, affinities, clones, cloneAffinities, nReplaces);
    end
    bestAffinity ← getBestAffinity(antibodies);
end
```

Since IT2FLC used in this study is type of fuzzy PD controller, it has two inputs and one output as Error (E), Error-Derivative (DE), and Force (F), respectively. The linguistic terms for these inputs and outputs are defined as in Table II. Here, LN, MN, Z, MP, and LP denote "Large Negative", "Medium Negative", "Zero", "Medium Positive", and "Large Positive", respectively. The total number of MFs to be optimized is 15. The fuzzy rules that map inputs and outputs are shown in Table III.

TABLE II. LINGUISTIC TERMS.

E	DE	F
LN	LN	LN
MN	MN	MN
Z	Z	Z
MP	MP	MP
LP	LP	LP

TABLE III. FUZZY RULE MATRIX FOR IT2FLC.

F		DE				
		LN	MN	Z	MP	LP
E	LN	LN	LN	LN	MN	Z
	MN	LN	MN	MN	Z	MP
	Z	MN	MN	Z	MP	MP
	MP	MN	Z	MP	MP	LP
	LP	Z	MP	LP	LP	LP

The first step in designing optimized MFs is to determine the input and output ranges of MFs. In [32]–[36], the domain of the inputs and the outputs of IT2FLC are divided into equal intervals and MFs are generated in these intervals. However, this method is not effective. In this way, the linguistic terms of IT2FLC can only be defined by MFs whose boundaries are formed in their corresponding intervals. In this case, MFs for linguistic terms that will bring the system to equilibrium faster against the actual inputs of the system cannot be generated in the region where they should be. For this reason, we create two IT2FSs named "*MinFSs*" and "*MaxFSs*" to define the ranges of the MFs. The MFs are generated between the ranges of the corresponding MFs defined in the domains of each input and output of these two IT2FSs. Figures 5 and 6 show the MFs of input "E" in the "*MinFSs*" and "*MaxFSs*" as an example, respectively. Similarly, other IT2FSs are created to define the MFs boundaries of other inputs and outputs in IT2FLC. We encode the MFs generated in their corresponding ranges into the triploid chromosomes. Thus, it is aimed at generating continued MFs in the whole domain of inputs or outputs of the IT2FLC.

In [32]–[36], MFs are generated in their corresponding intervals and encoded in triploid chromosomes for all linguistic terms mentioned in all IT2FLC rules. However, in an IT2FLC, a linguistic term defined by MFs can be used in more than one rule. In this case, multiple MFs are generated for the same linguistic term and encoded in the triploid chromosome. This causes the last generated MF for the same linguistic term to be taken into account, and the others are ignored when evaluating the affinity of this triploid chromosome. In later generations of the algorithm, RCQEA can generate a more effective MF for the performance of IT2FLC by mutation and crossover operations for the same linguistic term, but similarly, the last generated MF overwhelms all of the other MFs for the same linguistic term when evaluating the chromosome. Therefore, in this study, instead of encoding all MFs generated by inference from all linguistic terms mentioned in all the rules, one MF is generated and encoded in the triploid chromosomes for each linguistic term of IT2FLC. In this way, the algorithm runs more effectively for IT2FLC optimization.

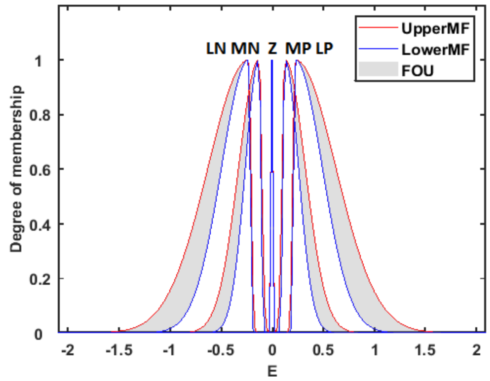

Fig. 5. The MFs of input “E” in the “MinFSs” IT2FSs.

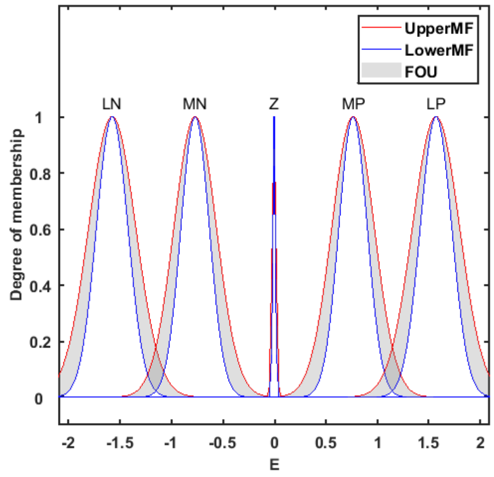

Fig. 6. The MFs of input “E” in the “MaxFSs” IT2FSs.

In [32]–[36], the input and output domains in IT2FLC are divided into equal-sized regions and Gaussian-type MFs are generated as shown in Fig. 1 for linguistic terms in these areas, as explained above. Unfortunately, in a continuous IT2FSs created with these Gaussian-type MFs generated in their corresponding intervals, the regions of the peak points of the MFs that have maximum membership degree can only be in limited areas. Since the Gaussian MF is symmetrical, when the mean value of the Gaussian-type MF to be created is determined at points outside the middle of the region reserved for it, the boundaries of the MF may exceed the range allocated to it and even the boundaries of other MFs around it. In this case, a possible more effective point for IT2FLC in the interval may not have a chance to be the mean value of Gaussian-type MF. Therefore, in this study, a skewed Gaussian function, namely Gauss2mf as in Fig. 7 is used for the membership functions. In this way, the most effective point for the IT2FLC can be the mean variable of the left- or right-skewed Gaussian MF and the MF does not exceed the limits allocated to it.

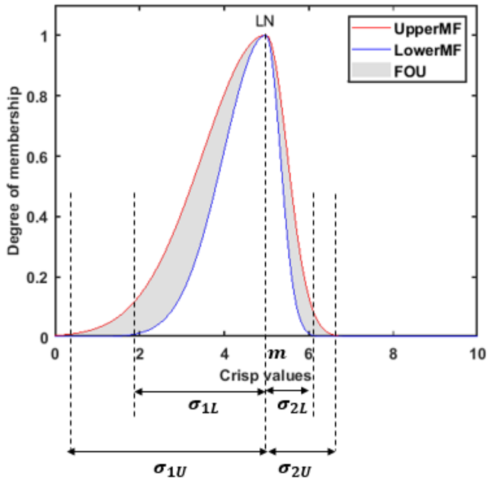

Fig. 7. An interval left skewed distribution type-2 MF.

A. Encoding

To optimize the MFs with RCQCSA, it is necessary to encode all the generated MFs into real-coded triploid chromosomes. In this study, there are 15 MFs in IT2FLC to control the inverted pendulum. To encode each generated MF to the triploid chromosome, the center point of the MF m, the width of the upper left MF σ_{1U}, the width of the lower left MF σ_{1L}, the width of the upper right MF σ_{2U}, and the width of the lower right MF σ_{2L} are used. Each triploid chromosome contains a pair of probability amplitudes of one qubit. In this study, 50 real-coded triploid chromosomes are generated for both RCQEA and RCQCSA to optimize IT2FLC. The coding scheme is shown in Table IV. The pole angle of the inverted pendulum system is used in the evaluation process of a chromosome. The fitness function in (19) is used to evaluate the performance of a chromosome by decoding it to the IT2FLC and getting the pole angle as the response of the inverted pendulum system

$$F = \sum_{t=1}^{200} |\theta(t)|. \tag{19}$$

Here, the simulation step size is determined as 0.01 and the system is simulated in 5 seconds.

TABLE IV. CHROMOSOME ENCODING SCHEMA.

m_1	σ_{11U}	σ_{11L}	σ_{12U}	σ_{12L}	**MF-1**
m_2	σ_{21U}	σ_{21L}	σ_{22U}	σ_{22L}	**MF-2**
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
m_{15}	σ_{151U}	σ_{151L}	σ_{152U}	σ_{152L}	**MF-15**

B. Discrete Crossover

Since the proposed RCQCSA is based on clonal selection principles, this algorithm does not use the discrete crossover operator. In the optimization of IT2FLC with RCQEA, unlike the studies in [32]–[36], if the continuity of the MFs in the domain is not ensured after the crossover process, the boundaries of MFs are rearranged using (20) and (21):

$$\sigma_{1U} = Min_{left} + r*(Max_{left} - Min_{left}), \tag{20}$$

$$\sigma_{2U} = Min_{right} + r*(Max_{right} - Min_{right}). \tag{21}$$

C. Mutation

In [32]–[36], mutation operations are performed on randomly selected genes that represent all MFs of linguistic terms in a rule from a randomly selected chromosome. The mean value of a Gaussian-type MF to be mutated is determined by using (14) and (15). The MF boundaries in the mutated gene are determined using (22)

$$W^t = \begin{cases} r*(Max - m^{t+1}), \, if \;\; m^{t+1} > (Max + Min)/2, \\ r*(m^{t+1} - Min), \, otherwise. \end{cases} \tag{22}$$

If the new IT2FLC designed with these mutated MFs is more effective than the old IT2FLC, the probability amplitudes are fixed; otherwise the probability amplitudes are updated by QRG as expressed in (17) and the rotation angle is defined in (18).

In this study, mutation operations are performed on randomly selected genes that represent a generated MF from a randomly selected chromosome. The right and left boundaries of the upper and lower MFs to be mutated are determined using (20) and (21).

The mean value of a skewed Gaussian-type MF to be mutated is determined by using (14) and (15) between this left and right boundaries. There may be other MFs to the right or left of the mutated MF in the domain to which the mutated MF belongs. In this case, the right and left upper MF boundaries of the mutated MF are revised again using (20) and (21) according to the boundaries of other MFs and the continuity of the MFs is ensured unlike previous studies. If the continuity of the MFs is not ensured, the fuzzification block cannot produce crisp output values for each input value in the domain to which the MFs belong.

D. Qrg

As briefly explained in Section IV, in case of generating a worse solution after the mutation operation of the RCQEA algorithm, quantum rotation is performed on the qubit with the QRG specified in (17). The rotation angle of QRG is determined as in (18). Figure 8 shows how this rotation should be for a state of a qubit. The purpose of the state of a qubit rotation operation, as explained in [23], [24], is to decrease $|\alpha_i^t|$ gradually and to increase $|\beta_i^t|$ gradually, and accordingly, to realize "Fine Search" in the neighborhood of the current solution and to realize "Coarse Search" in the whole solution space as the generation number of the algorithm increases.

Unfortunately, in some cases of probability amplitudes the rotation angle used to rotate the state of a qubit, determined by (18), reaches high magnitudes. As stated in [47], as an example, if the probability amplitudes are $\alpha = 0.01$ and $\beta = 0.99995$ (satisfying (13)), (18) produces a value of high magnitude for a rotation angle in excess of 2.0e7, while it can be $\pi/2$ at maximum. This causes the qubit to be rotated meaninglessly and makes it difficult for RCQEA to find better solutions. As the main contribution of this study, we propose a different rotation angle determination strategy as in (23)–(26) to solve this problem:

$$A = -\left(\frac{(sgn(\alpha_i^t \beta_i^t) - 1}{2}\right)\left(\frac{\pi}{2} - \theta_i^t\left(mod\,\frac{\pi}{2}\right)\right) sgn(\alpha_i^t \beta_i^t), \tag{23}$$

$$B = -\left(\frac{2(sgn(\alpha_i^t \beta_i^t) + 1)}{2}\right)\left(\frac{2(sgn(\alpha_i^t \beta_i^t) - 1)}{2}\frac{\pi}{2}\right), \tag{24}$$

$$C = \left(\frac{-(sgn(\alpha_i^t \beta_i^t) + 1)}{2}\theta_i^t\left(mod\,\frac{\pi}{2}\right)\right) sgn(\alpha_i^t \beta_i^t), \tag{25}$$

$$\Delta\theta_i^t = \frac{\pi}{4}\begin{pmatrix}(|A+B+C| - min) \\ /(max - min)\end{pmatrix}\left(-sgn(\alpha_i^t \beta_i^t)^{sgn(\alpha_i^t \beta_i^t)}\right). \tag{26}$$

Here, it is taken into account that the sign of the product of the probability amplitudes of the qubit state can be equal to either (-1), 0 or 1. A, B, and C specified in (23)–(25) produce a non-zero value when the sign of the product of the probability amplitudes is equal to (-1), 0, and 1, respectively.

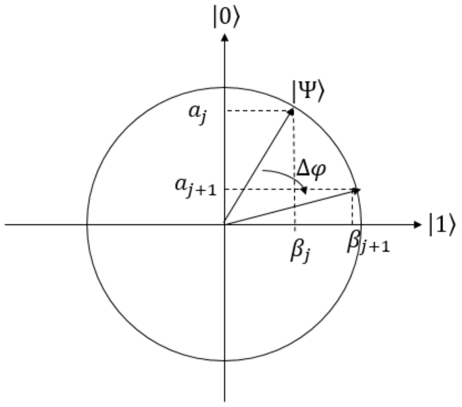

Fig. 8. Expended rotation of a qubits state.

The rotation angle is normalized between the *max* and *min* values specified in radians. Therefore, the state of qubits rotates with a larger angle if it is closer to the vertical axis than others. By using this proposed rotation angle determination technique, the state of the qubit rotates as desired in all cases, and RCQEA and the RCQCSA proposed in this study to optimize IT2FLC run more effectively.

VII. EXPERIMENTAL RESULTS

In this study, simulations are performed in MATLAB/SIMULINK environment. The performance analysis of the two algorithms is carried out in our study, where we propose the RCQCSA algorithm based on the clonal selection principles, together with the changes we made in the mutation, crossover operators, and QRG of the RCQEA algorithm existing in the literature for the IT2FLC optimization. To observe the effect of our modifications on the IT2FLC optimization, 4 different analyses are obtained as before and after the modifications in both algorithms. To clear the confusion, the algorithms are named “RCQEA”, “RCQCSA”, “RCQEA” (Modified Crossover, Mutation and QRG) and “RCQCSA” (Modified Mutation and QRG).

In the inverted pendulum stabilization problem, the task of the controller is to vertically stabilize the inverted pendulum that is standing with a random pole angle at initial. Figures 9 and 10 show, respectively, the angle change of the pole and the position change of the cart to which the pole is hinged in the inverted pendulum system which is controlled by the IT2FLC that formed from the best solutions of the RCQEA, RCQCSA, RCQEA (Modified Crossover, Mutation and QRG), and RCQCSA (Modified Mutation and QRG) algorithms. According to these results, RCQEA (Modified Crossover, Mutation and QRG) and RCQCSA ((Modified Mutation and QRG) algorithms for IT2FLC optimization find the same final solution that brings the pendulum to the vertical position earlier and keeps it fixed on the vertical axis, and changes the position of the cart less. The angular velocity values of the pole may be large because the angle of the inverted pendulum changes rapidly, but the optimized IT2FLC can compensate for it.

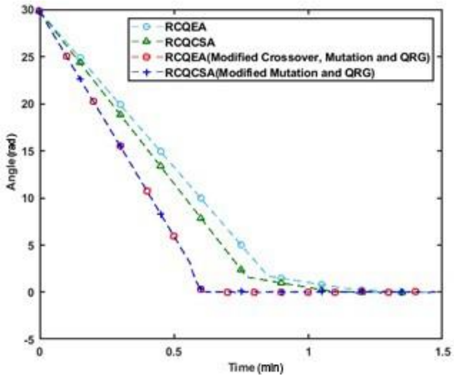

Fig. 9. Performance comparison on change of pole angles in inverted pendulum system controlled with IT2FLC that is optimized separately by RCQEA, RCQCSA, RCQEA (Modified Crossover and Mutation Gates) and RCQCSA (Modified Mutation Gate).

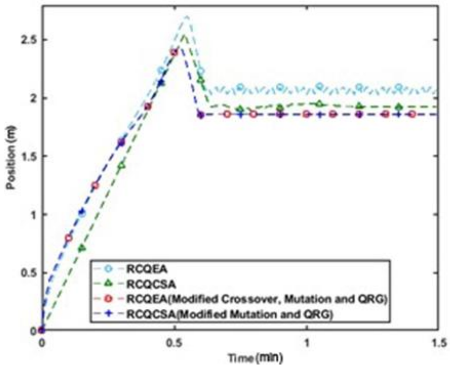

Fig. 10. Performance comparison on change of position in inverted pendulum system controlled with IT2FLC that is optimized separately by RCQEA, RCQCSA, RCQEA (Modified Crossover and Mutation Gates) and RCQCSA (Modified Mutation Gate).

Figure 11 shows the cost versus generation of IT2FLC which is designed by using the best solutions of the algorithms in the inverted pendulum system. According to Fig. 11, the proposed RCQCSA (Modified Mutation and QRG) for IT2FLC optimization finds the optimal solution in fewer generations, thus the convergence speed of the RCQEA algorithm in the literature is accelerated. It is also seen in Figs. 9–11 that the best solutions found by RCQEA (Modified Crossover, Mutation and QRG) and RCQCSA (Modified Mutation and QRG) in the latest generation are the same.

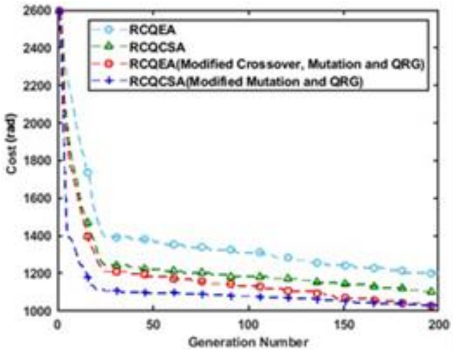

Fig. 11. Cost comparison of IT2FLCs optimized by RCQEA, RCQCSA, RCQEA (Modified Crossover and Mutation Gates) and RCQCSA (Modified Mutation Gate) algorithms on stabilizing the inverted pendulum system.

Figures 12–14 show the optimized continued MFs for “E”, “DE”, and “F”, respectively. As can be seen in Fig. 12, MF for the linguistic term “LP” has a mean value of 0.52, which is the initial pole angle of the inverted pendulum system. Thus, the stability of the inverted pendulum system is ensured by optimizing the MFs without optimizing the rules of IT2FLC.

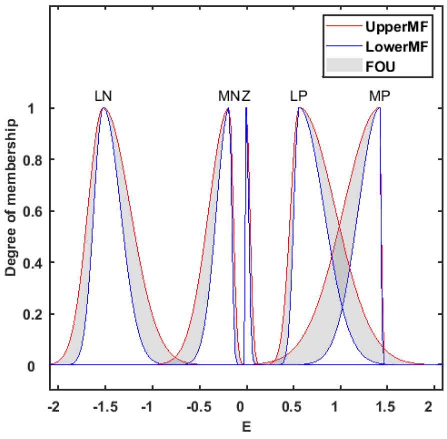

Fig. 12. Optimized MFs for the input “E”.

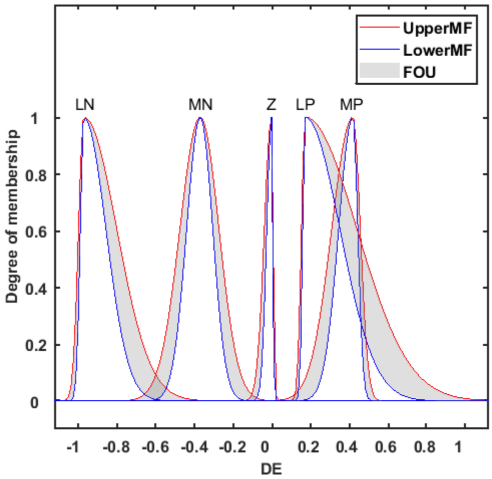

Fig. 13. Optimized MFs for the input “DE”.

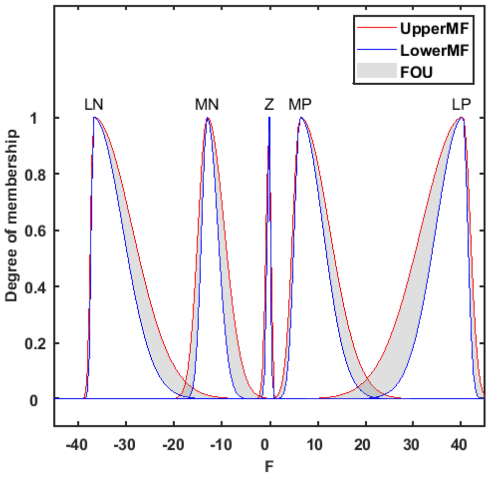

Fig. 14. Optimized MFs for the output "F".

Table V shows the integral absolute error (IAE) and settling time of the angle response of the system controlled by IT2FLC that is optimized with algorithms with different rotation angle determination strategies in QRG. The qubit is rotated $\pi/100$, $\pi/250$, $pi/20+rand+pi/20$ step size in [21], [47], and [56], respectively. According to the table, our proposed rotation angle update strategy in QRG of the algorithm allows the algorithm to find the appropriate solutions in the solution space. The minimum IAE and the settling time of the system's angle response show that the optimization algorithms using proposed QRG and modified mutation and crossover converge faster than others.

TABLE V. SYSTEM RESPONSE WITH DIFFERENT OPTIMIZATION ALGORITHMS USING DIFFERENT QUANTUM ROTATION GATES.

QRG	Algorithm	IAE (rad)	Settling Time (s)
Rotation angle determined in [21]	RCQEA	1.368E + 3	1.10
	RCQCSA	1.29E + 3	1.04
Rotation angle determined in [47]	RCQEA	1.258E + 3	0.854
	RCQCSA	1.23E + 3	0.82
Rotation angle determined in [56]	RCQEA	1.178E + 3	0.70
	RCQCSA	1.162E + 3	0.682
Our proposed rotation angle update strategy	RCQEA	1.136E + 3	0.64
	RCQCSA	1.136E + 3	0.64

VIII. CONCLUSIONS

In this study, improvements are realized in the RCQEA algorithm existing in the literature. A different strategy is proposed to determine the rotation angle of the qubit by modifying the QRG of the algorithm. The RCQEA algorithm is tested on IT2FLC optimization by modifying its QRG, mutation and crossover gates. Additionally, in this study, the RCQCSA algorithm based on clonal selection principles using modified mutation and QRG operations is proposed for IT2FLC optimization. To observe the performance of IT2FLCs response, the inverted pendulum stabilization problem, which is quite popular in the literature, is chosen. In addition, the mistakes and deficiencies in the studies existing in the literature on IT2FLC optimization with the RCQEA algorithm are explained and solutions are produced. The results show that the RCQCSA algorithm using the modified and improved mutation gate and QRG has a faster convergence speed in IT2FLC optimization. In future studies, it is aimed to apply the proposed method to different optimization problems, such as function optimization or optimization of controller of a real inverted pendulum system.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

[1] L. A. Zadeh, "Fuzzy Sets", *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965. DOI: 10.1016/S0019-9958(65)90241-X.

[2] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning—I", *Information Sciences*, vol. 8, no. 3, pp. 199–249, 1975. DOI: 10.1016/0020-0255(75)90036-5.

[3] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning—II", *Information Sciences*, vol. 8, no. 4, pp. 301–357, 1975. DOI: 10.1016/0020-0255(75)90046-8.

[4] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning-III", *Information Sciences*, vol. 9, no. 1, pp. 43–80, 1975. DOI: 10.1016/0020-0255(75)90017-1.

[5] J. M. Mendel, "Uncertainty, fuzzy logic, and signal processing", *Signal Processing*, vol. 80, no. 6, pp. 913–933, 2000. DOI: 10.1016/S0165-1684(00)00011-6.

[6] N. N. Karnik and J. M. Mendel, "Operations on type-2 fuzzy sets", *Fuzzy sets Syst.*, vol. 122, no. 2, pp. 327–348, 2001. DOI: 10.1016/S0165-0114(00)00079-8.

[7] N. N. Karnik and J. M. Mendel, "Centroid of a type-2 fuzzy set", *Inf. Sci.*, vol. 132, nos. 1–4, pp. 195–220, 2001. DOI: 10.1016/S0020-0255(01)00069-X.

[8] S. M. M. Golsefid, M. H. F. Zarandi, and I. B. Turksen, "Multi-central general type-2 fuzzy clustering approach for pattern recognitions", *Information Sciences*, vol. 328, pp. 172–188, 2016. DOI: 10.1016/j.ins.2015.08.027.

[9] D. Andriukaitis, A. Laucka, A. Valinevicius, M. Zilys, V. Markevicius, D. Navikas, R. Sotner, J. Petrzela, J. Jerabek, N. Herencsar, and D. Klimenta, "Research of the Operator's Advisory System Based on Fuzzy Logic for Pelletizing Equipment," *Symmetry*, vol. 11, no. 11, p. 1396, Nov. 2019. DOI: 10.3390/sym11111396.

[10] M. Antonelli, D. Bernardo, H. Hagras, and F. Marcelloni, "Multiobjective evolutionary optimization of type-2 fuzzy rule-based systems for financial data classification", *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 2, pp. 249–264, 2017. DOI: 10.1109/TFUZZ.2016.2578341.

[11] C. I. Gonzalez, P. Melin, J. R. Castro, O. Mendoza, and O. Castillo, "An improved sobel edge detection method based on generalized type-2 fuzzy logic", *Soft Comput.*, vol. 20, no. 2, pp. 773–784, 2016. DOI: 10.1007/s00500-014-1541-0.

[12] H. A. Hagras, "A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots", *IEEE Trans. Fuzzy Syst.*, vol. 12, no. 4, pp. 524–539, 2004. DOI: 10.1109/TFUZZ.2004.832538.

[13] T. Kumbasar, I. Eksin, M. Guzelkaya, and E. Yesil, "Type-2 fuzzy model based controller design for neutralization processes", *ISA Trans.*, vol. 51, no. 2, pp. 277–287, 2012. DOI: 10.1016/j.isatra.2011.10.007.

[14] D. Wu and W. W. Tan, "Interval type-2 fuzzy PI controllers: Why they are more robust", in *Proc. of 2010 IEEE International Conference on Granular Computing*, 2010, pp. 802–807. DOI: 10.1109/GrC.2010.112.

[15] M. Aliasghary, İ. Eksin, M. Güzelkaya, and T. Kumbasar, "Design of an interval type-2 fuzzy logic controller based on conventional PI controller", in *Proc. of 2012 20th Mediterranean Conference on Control & Automation (MED)*, 2012, pp. 627–632. DOI: 10.1109/MED.2012.6265708.

[16] D. Dasgupta (ed.), *Artificial Immune Systems and Their Applications*. Springer, 2012.

[17] J. Timmis, A. Hone, T. Stibor, and E. Clark, "Theoretical advances in artificial immune systems", *Theor. Comput. Sci.*, vol. 403, no. 1, pp. 11–32, 2008. DOI: 10.1016/j.tcs.2008.02.011.

[18] N. K. Jerne, "The immune system", *Sci. Am.*, vol. 229, no. 1, pp. 52–63, 1973. DOI: 10.1038/scientificamerican0773-52.

[19] G. L. Ada and S. G. Nossal, "The clonal-selection theory", *Sci. Am.*, vol. 257, no. 2, pp. 62–69, 1987. DOI:

10.1038/scientificamerican0887-62.

[20] L. N. de Castro and F. J. Von Zuben, "Learning and optimization using the clonal selection principle", *IEEE Trans. Evol. Comput.*, vol. 6, no. 3, pp. 239–251, 2002. DOI: 10.1109/TEVC.2002.1011539.

[21] K.-H. Han and J.-H. Kim, "Quantum-inspired evolutionary algorithm for a class of combinatorial optimization", *IEEE Trans. Evol. Comput.*, vol. 6, no. 6, pp. 580–593, 2002. DOI: 10.1109/TEVC.2002.804320.

[22] A. Narayanan and M. Moore, "Quantum-inspired genetic algorithms", in *Proc. of IEEE International Conference on Evolutionary Computation*, 1996, pp. 61–66. DOI: 10.1109/ICEC.1996.542334.

[23] R. Zhang and H. Gao, "Real-coded quantum evolutionary algorithm for complex functions with high-dimension", in *Proc. of 2007 International Conference on Mechatronics and Automation*, 2007, pp. 2974–2979. DOI: 10.1109/ICMA.2007.4304033.

[24] H. Gao and R. Zhang, "Real-coded quantum evolutionary algorithm for global numerical optimization with continuous variables", *Chinese J. Electron.*, vol. 20, no. 3, pp. 499–503, 2011.

[25] T. Zhao, Y. Xiang, S. Dian, R. Guo, and S. Li, "Hierarchical interval type-2 fuzzy path planning based on genetic optimization", *J. Intell. Fuzzy Syst.*, vol. 39, no. 1, pp. 937–948, 2020. DOI: 10.3233/JIFS-191864.

[26] K. M. Sreedivya, P. A. Jeyanthy, and D. Devaraj, "Improved design of interval type-2 fuzzy based wide area power system stabilizer for inter-area oscillation damping", *Microprocessors and Microsystems*, vol. 83, art. 103957, 2021. DOI: 10.1016/j.micpro.2021.103957.

[27] J. Wang and T. Kumbasar, "Parameter optimization of interval Type-2 fuzzy neural networks based on PSO and BBBC methods", *IEEE/CAA J. Autom. Sin.*, vol. 6, no. 1, pp. 247–257, 2019. DOI: 10.1109/JAS.2019.1911348.

[28] C.-J. Lin, S.-Y. Jeng, H.-Y. Lin, and C.-Y. Yu, "Design and verification of an interval type-2 fuzzy neural network based on improved particle swarm optimization", *Appl. Sci.*, vol. 10, no. 9, p. 3041, 2020. DOI: 10.3390/app10093041.

[29] D. S. Mai, T. H. Dang, and L. T. Ngo, "Optimization of interval type-2 fuzzy system using the PSO technique for predictive problems", *Journal of Information and Telecommunication*, vol. 5, no. 2, pp. 197–213, 2021. DOI: 10.1080/24751839.2020.1833141.

[30] F. Valdez and C. Peraza, "Dynamic parameter adaptation in the harmony search algorithm for the optimization of interval type-2 fuzzy logic controllers", *Soft Comput.*, vol. 24, no. 1, pp. 179–192, 2020. DOI: 10.1007/s00500-019-04124-x.

[31] A. Ahmadi, H. Abdul Rahim, and R. Abdul Rahim, "Optimization of a self-tuning PID type fuzzy controller and a PID controller for an inverted pendulum", *Journal of Intelligent & Fuzzy Systems*, vol. 26, no. 4, pp. 1987–1999, 2014. DOI: 10.3233/IFS-130877.

[32] P. C. Shill, Md. F. Amin, M. A. H. Akhand, and K. Murase, "Optimization of interval type-2 fuzzy logic controller using quantum genetic algorithms", in *Proc. of 2012 IEEE International Conference on Fuzzy Systems*, 2012, pp. 1–8. DOI: 10.1109/FUZZ-IEEE.2012.6251207.

[33] P. C. Shill, Md. A. Hossain, Md. K. Hossain, Md. F. Amin, and K. Murase, "Design and implementation of an effective fuzzy logic controller based on quantum inspired evolutionary algorithm", *J. Comput.*, vol. 7, no. 3, pp. 586–596, 2012. DOI: 10.4304/jcp.7.3.586-596.

[34] K. Murase, P. C. Shill, B. C. Sarker, and M. C. Urmi, "Quantum fuzzy controller for inverted pendulum system based on quantum genetic optimization", *Int. J. Adv. Res. Comput. Sci.*, vol. 3, no. 7, 2012.

[35] P. C. Shill, M. A. H. Akhand, and K. Murase, "Fuzzy logic controller for an inverted pendulum system using quantum genetic optimization", in *Proc. of 14th International Conference on Computer and Information Technology (ICCIT 2011)*, 2011, pp. 503–508. DOI: 10.1109/ICCITechn.2011.6164841.

[36] P. C. Shill, Md. A. Hossain, Md. F. Amin, and K. Murase, "An adaptive fuzzy logic controller based on real coded quantum-inspired evolutionary algorithm", in *Proc. of 2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011)*, 2011, pp. 614–621. DOI: 10.1109/FUZZY.2011.6007374.

[37] E. Bernal, M. L. Lagunes, O. Castillo, J. Soria, and F. Valdez, "Optimization of type-2 fuzzy logic controller design using the GSO and FA algorithms", *International Journal of Fuzzy Systems*, vol. 23, no. 1, pp. 42–57, 2021. DOI: 10.1007/s40815-020-00976-w.

[38] M. Nikolić, M. Šelmić, D. Macura, and J. Ćalić, "Bee colony optimization metaheuristic for fuzzy membership functions tuning", *Expert Systems with Applications*, vol. 158, art. 113601, 2020. DOI: 10.1016/j.eswa.2020.113601.

[39] R.-E. Precup, R.-C. David, R.-C. Roman, A.-I. Szedlak-Stinean, and E. M. Petriu, "Optimal tuning of interval type-2 fuzzy controllers for nonlinear servo systems using Slime Mould Algorithm", *International Journal of Systems Science*, pp. 1–16, 2021. DOI: 10.1080/00207721.2021.1927236.

[40] F. Cuevas, O. Castillo, and P. Cortes-Antonio, "Optimal design of interval type-2 fuzzy tracking controllers of mobile robots using a metaheuristic algorithm", in *Recent Advances of Hybrid Intelligent Systems Based on Soft Computing. Studies in Computational Intelligence*, vol. 915. Springer, Cham, 2021, pp. 315–341. DOI: 10.1007/978-3-030-58728-4_18.

[41] C. A. V Guerrero, P. M. Silveira, and J. M. C. Filho, "Adaptation of the clonal selection algorithm to the real-time coordinated Volt/VAr control through a software-in-the-loop strategy", *Electr. Power Syst. Res.*, vol. 194, art. 107092, 2021. DOI: 10.1016/j.epsr.2021.107092.

[42] M. W. Hussain, B. Pradhan, X. Z. Gao, K. H. K. Reddy, and D. S. Roy, "Clonal selection algorithm for energy minimization in software defined networks", *Appl. Soft Comput.*, vol. 96, art. 106617, 2020. DOI: 10.1016/j.asoc.2020.106617.

[43] H. Siqueira, J. T. Belotti, L. Boccato, I. Luna, R. Attux, and C. Lyra, "Recursive linear models optimized by bioinspired metaheuristics to streamflow time series prediction", *Int. Trans. Oper. Res.*, 2020. DOI: 10.1111/itor.12908.

[44] A. M. Şakiroğlu and A. Arslan, "Optimization of fuzzy membership function using clonal selection", in *Adaptive and Natural Computing Algorithms. ICANNGA 2007. Lecture Notes in Computer Science*, vol. 4431. Springer, Berlin, Heidelberg, 2007, pp. 694–701. DOI: 10.1007/978-3-540-71618-1_77.

[45] A. M. Acilar and A. Arslan, "Optimization of multiple input single output fuzzy membership functions using clonal selection algorithm", *Expert Systems with Applications*, vol. 38, no. 3, pp. 1374–1381, 2011. DOI: 10.1016/j.eswa.2010.07.036.

[46] H. Wang, J. Liu, J. Zhi, and C. Fu, "The improvement of quantum genetic algorithm and its application on function optimization", *Math. Probl. Eng.*, vol. 2013, art. ID 730749, 2013. DOI: 10.1155/2013/730749.

[47] J. Wright and I. Jordanov, "Convergence properties of quantum evolutionary algorithms on high dimension problems", *Neurocomputing*, vols. 326–327, pp. 82–99, 2019. DOI: 10.1016/j.neucom.2017.08.065.

[48] J. Wright and I. Jordanov, "Quantum inspired evolutionary algorithms with improved rotation gates for real-coded synthetic and real world optimization problems", *Integr. Comput. Aided. Eng.*, vol. 24, no. 3, pp. 203–223, 2017. DOI: 10.3233/ICA-170545.

[49] Md. A. Hossain, Md. K. Hossain, and M. M. A. Hashem, "Hybrid real-coded quantum evolutionary algorithm based on particle swarm theory", in *Proc. of 2009 12th International Conference on Computers and Information Technology*, 2009, pp. 13–18. DOI: 10.1109/ICCIT.2009.5407175.

[50] H. Dai, Y. Yang, H. Li, and C. Li, "Bi-direction quantum crossover-based clonal selection algorithm and its applications", *Expert Syst. Appl.*, vol. 41, no. 16, pp. 7248–7258, 2014. DOI: 10.1016/j.eswa.2014.05.053.

[51] R. Shang, L. Jiao, Y. Ren, L. Li, and L. Wang, "Quantum immune clonal coevolutionary algorithm for dynamic multiobjective optimization", *Soft Comput.*, vol. 18, no. 4, pp. 743–756, 2014. DOI: 10.1007/s00500-013-1085-8.

[52] X. Cai *et al.*, "An improved quantum-inspired cooperative co-evolution algorithm with muli-strategy and its application", *Expert Syst. Appl.*, vol. 171, art. 114629, 2021. DOI: 10.1016/j.eswa.2021.114629.

[53] X. Xu, J. Jiang, J. Jie, H. Wang, and W. Wang, "An improved real coded quantum genetic algorithm and its applications", in *Proc. of 2010 International Conference on Computational Aspects of Social Networks*, 2010, pp. 307–310. DOI: 10.1109/CASoN.2010.75.

[54] Y.-X. Zhang, X.-Y. Qian, H.-D. Peng, and J.-H. Wang, "An allele real-coded quantum evolutionary algorithm based on hybrid updating strategy", *Computational Intelligence and Neuroscience*, vol. 2016, art. ID 9891382, 2016. DOI: 10.1155/2016/9891382.

[55] W. Deng, J. Xu, H. Zhao, and Y. Song, "A novel gate resource allocation method using improved PSO-based QEA", *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 3, pp. 1737–1745, 2022. DOI: 10.1109/TITS.2020.3025796.

[56] H. Talbi and A. Draa, "A new real-coded quantum-inspired evolutionary algorithm for continuous optimization", *Applied Soft Computing*, vol. 61, pp. 765–791, 2017. DOI: 10.1016/j.asoc.2017.07.046.

[57] J. M. Mendel and R. I. B. John, "Type-2 fuzzy sets made simple", *IEEE Trans. fuzzy Syst.*, vol. 10, no. 2, pp. 117–127, 2002. DOI: 10.1109/91.995115.

[58] Q. Liang and J. M. Mendel, “Interval type-2 fuzzy logic systems: Theory and design”, *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 5, pp. 535–550, 2000. DOI: 10.1109/91.873577.

[59] D. Wu and J. M. Mendel, “Designing practical interval type-2 fuzzy logic systems made simple”, in *Proc. of 2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2014, pp. 800–807. DOI: 10.1109/FUZZ-IEEE.2014.6891534.

[60] K. Mittal, A. Jain, K. S. Vaisla, O. Castillo, and J. Kacprzyk, “A comprehensive review on type 2 fuzzy logic applications: Past, present and future”, *Eng. Appl. Artif. Intell.*, vol. 95, art. 103916, 2020. DOI: 10.1016/j.engappai.2020.103916.

[61] J. M. Mendel, *Untercain Rule-Based Fuzzy Systems: Introduction and New Directions*, 2nd ed. Springer Cham, 2017. DOI: 10.1007/978-3-319-51370-6.

[62] H.-X. Li and H. B. Gatland, “Conventional fuzzy control and its enhancement”, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 26, no. 5, pp. 791–797, 1996. DOI: 10.1109/3477.537321.

[63] A. K. Yadav, P. Gaur, A. P. Mittal, and M. Anzar, “Comparative analysis of various control techniques for inverted pendulum”, in *Proc. of India International Conference on Power Electronics 2010 (IICPE2010)*, 2011, pp. 1–6. DOI: 10.1109/IICPE.2011.5728071.

[64] G. Zhang, W. Jin, and L. Hu, “Quantum evolutionary algorithm for multi-objective optimization problems”, in *Proc. of the 2003 IEEE International Symposium on Intelligent Control*, 2003, pp. 703–708. DOI: 10.1109/ISIC.2003.1254721.

[65] K.-H. Han and J.-H. Kim, “Quantum-inspired evolutionary algorithms with a new termination criterion, H/sub/spl epsi//gate, and two-phase scheme”, *IEEE Trans. Evol. Comput.*, vol. 8, no. 2, pp. 156–169, 2004. DOI: 10.1109/TEVC.2004.823467.