Neural Comput & Applic (2017) 28:979–999
DOI 10.1007/s00521-015-2111-9

REVIEW

Recent advances on the use of meta-heuristic optimization algorithms to optimize the type-2 fuzzy logic systems in intelligent control

Mukhtar Fatihu Hamza[1,2] · Hwa Jen Yap[1] · Imtiaz Ahmed Choudhury[1]

Received: 14 May 2015 / Accepted: 4 November 2015 / Published online: 21 November 2015

Abstract Finding the appropriate values of parameters and structure of type-2 fuzzy logic systems is a difficult and complex task. Many types of meta-heuristic algorithms have been used to find the complex structure and appropriate parameter values of the type-2 fuzzy systems and more recently hybrid meta-heuristic algorithms. In this paper, we review recent advances (2012 to date) on the application of meta-heuristic algorithms and hybrid meta-heuristic algorithms, for the optimization of type-2 fuzzy logic systems in intelligent control. It was found that the major meta-heuristic algorithms used for optimizing the design of type-2 fuzzy logic systems in intelligent control were genetic algorithms and particle swarm optimization as well as hybrid meta-heuristic algorithms. Researchers can use this review as a starting point for further advancement as well as an exploration of other meta-heuristic algorithms that have received little or no attention from researchers.

Keywords Type-2 fuzzy logic systems · Intelligent control · Genetic algorithm · Particle swarm optimization · Hybrid meta-heuristic algorithms

✉ Hwa Jen Yap
hjyap737@um.edu.my

Mukhtar Fatihu Hamza
emukhtarfh@gmail.com

1 Department of Mechanical Engineering, University of Malaya, 50603 Kuala Lumpur, Malaysia

2 Department of Mechatronics Engineering, Bayero University, Kano 3011, Nigeria

1 Introduction

The data available in many real-world problems such as control, time series forecasting, pattern recognition, decision making, system identification and modeling are quite associated with uncertainties in nature [1]. This is due to deficiency in information which may be imprecise, incomplete, contradictory, vague, unreliable, fragmentary or deficient in some other way. Uncertainty is an inherent characteristic of an information [2]. The fuzzy logic theory increased the ability of systems to cope with the uncertainty problems. The basic feature of fuzzy reasoning allows for handling different kind of uncertainties [3]. In type-1 fuzzy, the uncertainty is represented by a precise number in a range of (0, 1) interpreted as a degree of membership function (MF). In view of the fact that it is too difficult to know a precise value for uncertainty, working with type-1 model is more reasonable. However, some researchers argued that in cases where there is high level of uncertainty, type-1 fuzzy has limited ability to handle it because its membership degree for each input is a crisp number [4]. The type-2 fuzzy logic system (T2FLS) which uses the type-2 fuzzy set (T2FS) was introduced to circumvent the limitations of the type-1 fuzzy [4–6]. The main characteristic of T2FLS is that, its MFs are fuzzy. Therefore, it has more degree of freedom in designing verities of systems with uncertainties [7–9]. The T2FLS is of two types, namely, interval type-2 fuzzy logic system (IT2FLS) that uses interval type-2 fuzzy sets (IT2FSs) and general type-2 fuzzy logic system (GT2FLS) that differ from IT2FSs because the GT2FLS uses general type-2 fuzzy sets (GT2FS) [10]. It was argued that in the presence of uncertainty, T2FLS is preferred over the type-1 fuzzy logic system (T1FLS) [11]. Similarly, there are several records of experimental evidences illustrating some significant improvements in terms of accuracy of T2FLS over T1FLS counterpart [12–14]. Despite the significance of T2FLS, there is no systematic and comprehensive methodology for the design of T2FLS [11]. However, using meta-heuristic algorithms to optimize the structure and parameter values of the FLS using IT2FSs in a FLS has the potential to provide better performance for a FLS than using type-1 fuzzy logic sets (T1FSs). Similarly, using GT2FSs in a FLS has the potential to provide better (and certainly not worst) performance for a FLS over IT2FSs. This is because IT2FLS has more design parameters (degree of freedom) than T1FLS. In addition, GT2FLS has more design parameters (degree of freedom) than IT2FLS [10].

Meta-heuristic optimization algorithms refer to a class of soft computing techniques that relate to the searching of optimal, satisfactory or best solution for a particular problem. The solution can be the absolute best out of other alternative solutions [15]. Meta-heuristic algorithms are found to be useful in many applications domain, [16] especially hybrid [17].

Generally, the design of type-2 fuzzy model based on experimental data can be categorized into two; the first is constructing the type-2 fuzzy model from the existing optimal type-1 fuzzy model, and the second is direct design of type-2 fuzzy model from the experimental data. In both of the methods, the manual design and tuning of the MFs of T2FLS to give a good response is a difficult task [18]. The use of meta-heuristic optimization algorithms can help in getting an optimal type-2 fuzzy model for a particular application [11]. The optimization of the MF parameters of a T2FLS was recommended to be at the design phase. Subsequently, the optimal parameters are fixed during the operation phase except in a case of continuous adaptation [10].

Recently, many meta-heuristic optimization algorithms such as genetic algorithm (GA), particle swarm optimization (PSO), ant colony optimization (ACO), big bang–big crunch optimization, bacterial foraging optimization, biogeography optimization, chemical optimization (CO), back propagation algorithm (BPA), simulated annealing, firefly algorithm FA, tabu search optimization, and hybrid optimization (HO) have been used for the design of a T2FLC for different control applications [15, 19–29].

There are reviews in the literature regarding the T2FLS for different applications. For example, in literature [30], the applications of T2FLS in classification, clustering, and pattern recognition were conducted. In a related study, T2FLS applications in pattern recognition and classification were reviewed in literature [31]. In a review presented in literature [32], the authors reviewed the design and optimization of interval type-2 fuzzy logic controller (IT2FLC). Furthermore, the review in literature [11] presents the general case of optimization of T2FLS using bio-inspired method.

The previous reviews mainly focused on the single meta-heuristic algorithms without HO. In addition, the reviews presented in [11, 30–32] mainly covered papers from the year 2000 to 2011. In the review paper of Castillo and Melin [18], a review on interval type-2 fuzzy logic applications in intelligent control was presented. The authors focused on related papers on interval type-2 fuzzy from 2004 to 2014. This is because at that moment, most of the type-2 fuzzy controllers considered in the application only used interval type-2 fuzzy sets as argued in that paper. The trend of this area of research has been extended to the use of HO methods. Moreover, recently the application of GT2FS in this area of research is increasing [27, 33–35]. These were not covered in the previous reviews conducted by researchers, and it is part of their recommendations.

In this paper, we present a review of the recent advances on the use of meta-heuristic optimization algorithms in optimizing the design of T2FLS in intelligent control. We attempt to conduct a comprehensive literature review of the studies conducted in this domain to provide a state-of-the-art review to prevent replication of what has already been accomplished. Additionally, we intend to provide a clear perspective with a broad and in-depth review of the research studies in this domain, and we intend for researchers to use this review as a starting point for further advancement as well as an exploration of other techniques that have received little or no attention from researchers. These stated objectives were the driving force that motivated this review article.

The contributions of the paper are summarized as follows:

- Recent advances from 2012 to date on the utilization of meta-heuristic optimization algorithms in optimization base design of T2FLS in intelligent control are summarized.
- Provides analysis and synthesis of the published articles from 2012 to date with insight on proposed algorithms, algorithms compared with, other controllers compared with type-2 fuzzy logic controller (T2FLC), and major findings are highlighted.
- Provides real pervasiveness of using meta-heuristic algorithms for the optimization of type 2 fuzzy logic controllers for engineering design benefiting to researchers intending to use meta-heuristic algorithms in the design of this type of controller.
- Open research problems and future research direction in the design of type 2 fuzzy logic controllers are highlighted for easy identification by researchers.

2 Type-2 fuzzy logic systems

The idea of fuzzy logic systems and T2FS was pioneered by Zadeh in 1965 and 1975, respectively [2, 3]. A concise overview of T2FLS was presented in this section with the intention of providing readers with the basic knowledge of how type-2 fuzzy operates to achieve its objective.

Imagine blurring the type-1 membership function (T1-MF) as shown in Fig. 1a by moving the points on the triangle from left to right. Type-2 membership function (T2-MF) was formed as shown in Fig. 1b.

Considering a specific value of x, say x', in T1-MF that has a specific crisp value μ. On the other hand, in blurred type-1 MF, it does not have a single value. Instead, the MF has many values with different weight at all the point where the vertical line intersects the blur. The amplitude distribution can be assigned to all these points. Doing this for all $x \in X$, a three-dimensional MF was created that characterized a T2FS [5, 36]. A T2FS $\tilde{A}$, is characterized by T2-MF $\mu_{\tilde{A}}(x, y)$, for $x \in X$ and $u \in J_x \subseteq [1, 0]$, that is,

$$\tilde{A} = \left\{ \left((x, y), \mu_{\tilde{A}}(x, y) \right) \mid \forall x \in X,\ \forall y \in J_x \subseteq [0, 1] \right\} \tag{1}$$

In which $0 \leq \mu_{\tilde{A}}(x, u) \leq 1$.

The primary membership of x can be represented as: $J_x \subseteq [0, 1]$ and the secondary set is $\mu_{\tilde{A}}(x, u)$ which is T1FS. Therefore, a T2-membership grade should lie between or be equal to 0 and 1 [8]. Each primary membership has its corresponding secondary membership (also lies in [1, 0]) that defines its possibilities. The uncertainty can be represented by foot print of uncertainty (FOU) region [9].

As stated earlier, the T2FLS are of two types, namely IT2FLS and GT2FLS. In the former, all the secondary grades of the IT2FS are equal to 1, and it is completely described by upper MF and lower MF (UMF and LMF) as shown in Fig. 2a, and in the latter, the GT2FS is three dimensional (3D) which can be decomposed into 3D MF of a GT2FS by using different kinds of cut. The most commonly used cuts are horizontal slices, vertical slices and wavy slices as shown in Fig. 2b–d, respectively. The detailed explanation and formulation of IT2FS can be found in [1, 2, 4–9, 36–39] and also the detailed explanation and formulation of GT2FS are presented in literatures [10, 40–47].

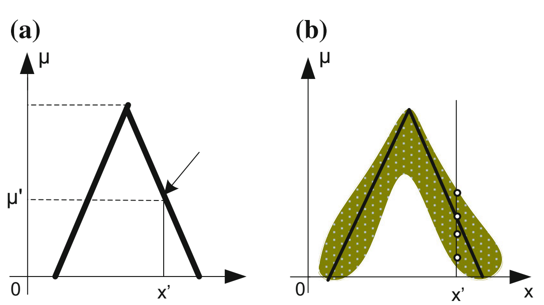

Fig. 1 **a** Type-1 MF, **b** blurred type-1 MF

Like type-1, the type-2 fuzzy is using IF–THEN rules, but its antecedent and consequent sets are type-2. Furthermore, type-2 consists of the following blocks; fuzzification, inference, and output processing. The output processing block comprises of the type reduction and the defuzzification blocks as shown in Fig. 3. For a better illustration, a fuzzy system with two crisp inputs and one crisp output is shown in Fig. 3, which is the same for both IT2FLS and GT2FLS. Each of the blocks is explain in brief in Sects. 2.1–2.4. For easy explanation, the interval type-2 fuzzy Mamdani system with center-of-set type reduction was used (IT2FS Takagi–Sugeno-King (TSK), GT2FS Mamdani and GT2FS TSK also exist) [48].

2.1 Fuzzification

The fuzzifier in type-1 fuzzy and type-2 fuzzy is doing the same work, which is transforming a numeric vector entries $X = (x_1 \ldots x_p)^{\mathrm{T}} \in X_1 \times X_1 \times \cdots \times X_p \equiv X$ into $\tilde{A}_x$ (type-2 fuzzy set) defined in X. Given the singleton numeric inputs, the mapping can be performed as follows [48]:

$$\begin{aligned} &\mu_{A_x}(x) = 1/1 \text{ with } X = X', \quad \text{and} \\ &\mu_{A_x}(x) = 1/0, \quad \text{for} \quad \forall X \in X \text{ with } x \neq x' \end{aligned} \tag{2}$$

Equation (2) shows that $\mu_{\tilde{X}_i}(x_i) = 1/1$ when $x_i = x_i'$ and $\mu_{\tilde{X}_i}(x_i) = 1/0$ when $x_i \neq x_i'$ for all $i = 1, \ldots, p$.

2.2 Rules

Both type-1 fuzzy and type-2 fuzzy use IF–THEN rules. In type-2, the antecedent and/or consequent MFs are represented by type-2 fuzzy sets. For T2FLC characterized by M rules with p inputs $x_1 \in X_1, \ldots, x_p \in X_p$ and one output $y \in Y$. The ith rule can be expressed as Eq. (3) [49].

$$R^i : \text{IF } x_1 \text{ is } \tilde{F}_1^i \text{ and} \ldots \text{and } x_p \text{ is } \tilde{F}_p^i, \quad \text{THEN } y \text{ is } \tilde{Y}^i \tag{3}$$

where $\tilde{F}_k^i (k = 1, \ldots, p)$ are type-2 antecedent fuzzy sets, $i = 1, \ldots, M$, and $\breve{Y}^i$ the output of the ith rule. The assumption here is that, all antecedent and consequent fuzzy set in Mamdani rules are type-2.

2.3 Inference

The inference mechanism in type-2 fuzzy is in the same as type-1 fuzzy. It is a rule combination to produce a mapping from some input type-2 fuzzy set to output type-2 fuzzy set. It is necessary to calculate the intersection, union and composition of type-2 relations in order to realize this mapping [50]. Equation (3) can be rewritten as:

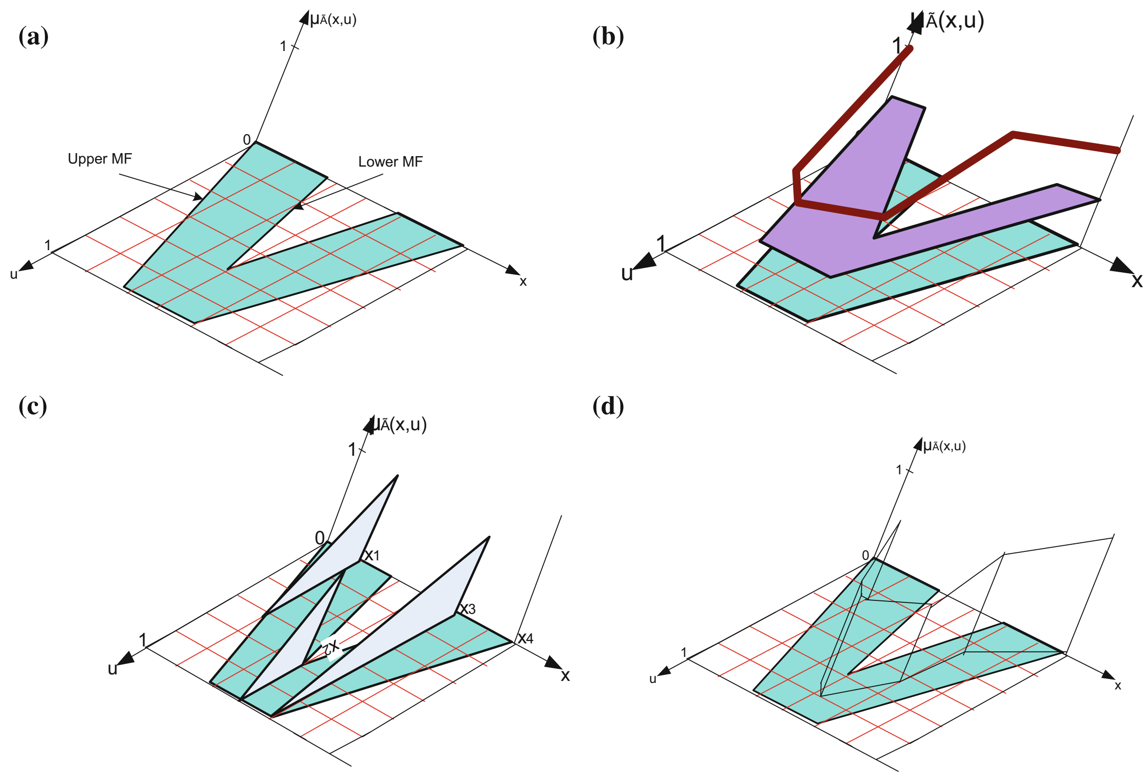

Fig. 2 Representation of different types of slices in GT2FS: **a** foot print of uncertainty, **b** *horizontal* slice, **c** *vertical* slice, **d** *wavy* slice

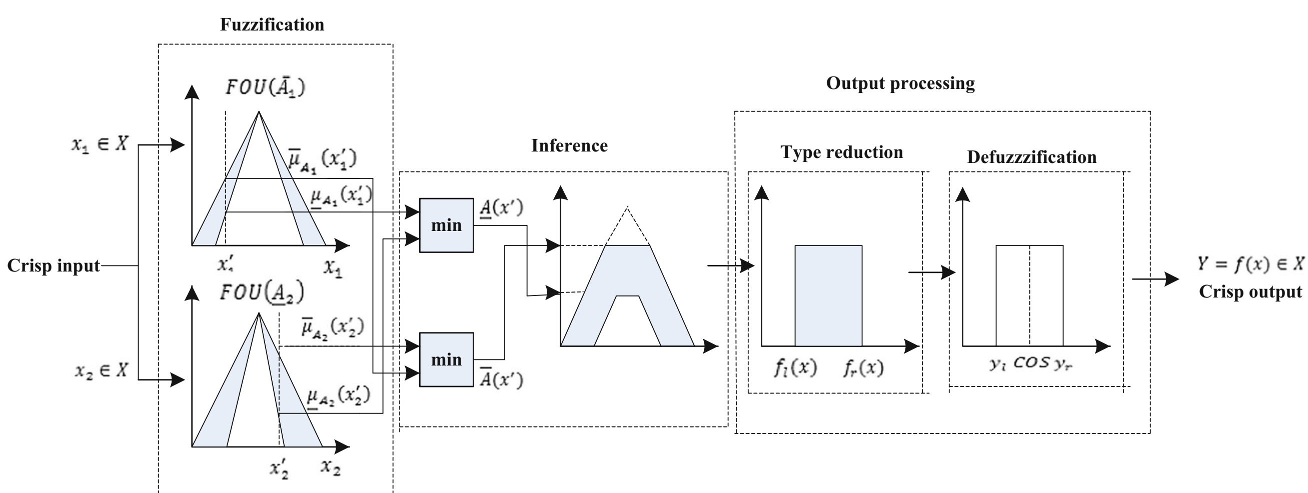

Fig. 3 Type-2 fuzzy logic system structure

$$R^i : \tilde{F}_1^i \times \cdots \times \tilde{F}_p^i \rightarrow \tilde{Y}^i = \tilde{A}^i \rightarrow \tilde{Y}^i, \quad i = 1, \ldots, M \tag{4}$$

R^i is described by the MF $\mu_{R^i}(X, y)$ for $X = (x_1, \ldots, x_p)$, where

$$\mu_{R^i}(X, y) = \mu_{\tilde{A}^i \rightarrow \tilde{Y}^i}(X, y) = \mu_{\tilde{F}_1^i}(x_1) \prod \cdots \prod_{\mu_{\tilde{F}_p^i}} (x_p) \prod_{\mu_{\tilde{Y}^i}} (y) = \left[\prod_{l=1}^{p} \mu_{\tilde{F}_l^i}(x_l) \right] \prod_{\mu_{\tilde{Y}^i}} (y) \tag{5}$$

Generally, the p-dimensional input to R^i is given by T2FS $\tilde{A}_x$ whose MF is

$$\mu_{\tilde{A}_x}(X) = \mu_{\tilde{x}_1}(x_1)\prod\cdots\prod\mu_{\tilde{x}_p}(x_p) = \prod_{l=1}^{p}\mu_{\tilde{x}_l}(x_l) \tag{6}$$

where $\tilde{X}_l(l = 1,\ldots,p)$ are the labels of the fuzzy sets describing the inputs. Each rule determines the T2FS $\tilde{B}^i = \tilde{A}_x \circ R^i$ such that:

$$\mu_{\tilde{B}^i}(y) = \mu_{\tilde{A}_x^\circ R^i} = \coprod_{x\in X}\mu_{\tilde{A}_x}(X)\prod\mu_{R^i}(X,y) \quad y\in G \quad i = 1,\ldots,M \tag{7}$$

This equation is the input–output relation in Fig. 3 between the T2FS that excite a single rule in the inference engine and the T2FS at the output of that engine.

The IT2FS was used in the FLS and meet under product t-norm. Therefore, the result of the input and antecedent operations which are contained in the firing set $\prod_{l=1}^{p}\mu_{F_{l_l}}(x'_l \equiv F^i(X'))$, is an interval type-1 fuzzy set [51]:

$$F^i(X') = \left[\underline{f}^i(X'), \bar{f}^i(X')\right] \equiv \left[\underline{f}^i, \bar{f}^i\right], \tag{8}$$

where

$$\underline{f}^i(X') = \underline{\mu}_{\tilde{F}_1^i}(x'_1)\times\cdots\times\underline{\mu}_{\tilde{F}_p^i}\left(x'_p\right) \tag{9}$$

and

$$\bar{f}^i(X') = \bar{\mu}_{F_1^i}(x'_1)\times\cdots\times\bar{\mu}_{F_p^i}\left(x'_p\right) \tag{10}$$

2.4 Output processing

The output processing constitutes the type reduction that generates the type-1 fuzzy set, and defuzzifier that converts the generated type-1 fuzzy set to the crisp output [52]. There are many kind of type reduction in literature, which includes center-of-sets, centroid, height, and modified height [53–55]. In this study, the center-of-set type reduction will be used for illustration as follows [7, 8].

$$Y_{\cos}\left(Y^1,\ldots,Y^M,F^1,\ldots,F^M\right) = [y_l, y_r] = \int_{y^1}\cdots\int_{y^M}\int_{f^1}\cdots\int_{f^M} 1\Bigg/\frac{\sum_{i=1}^{M}f^iy^i}{\sum_{i=1}^{M}f^i} \tag{11}$$

where $Y_{\cos}$ is the interval set determined by y_l and y_r (two end points); $f^i \in F^i = [\underline{f}^i, \bar{f}^i]$; $y^i \in Y^i = [\underline{y}^i, \bar{y}^i]$; $i = 1,\ldots,M$; and M is the number of rules.

It can be observed that each set on the right hand side of Eq. (11) is an interval type-1 set, hence the left hand of that equation $(Y_{\cos}(Y^1,\ldots,Y^M,F^1,\ldots,F^M))$ is also interval type-1 set. Therefore, $Y_{\cos}(Y^1,\ldots,Y^M,F^1,\ldots,F^M)$ can be found by just computing the y_l and y_r [7]. Karnik and Mendel [9, 54] have shown that y_l and y_r depend on maximum of $\underline{f}^i$ or $\bar{f}^i$ values as follows:

$$y_l = \frac{\sum_{i=1}^{M}f_l^iy_l^i}{\sum_{i=1}^{M}f_l^i}, \quad \text{and} \quad y_r = \frac{\sum_{i=1}^{M}f_r^iy_r^i}{\sum_{i=1}^{M}f_r^i} \tag{12}$$

where f_l^i and f_r^i denotes the firing strength membership grade (either $\underline{f}^i$ or $\bar{f}^i$) contributing to the left-most point y_l and right most point y_r, respectively.

The defuzzifier of an interval type-2 Fuzzy can be calculated as follows [7]:

$$y(x) = \frac{y_l + y_r}{2} \tag{13}$$

3 Fuzzy logic controllers

Most industrial systems are nonlinear in nature and exhibit some level of uncertainty [6, 56]. Some modern control was used in the past decade, such as nonlinear, adaptive, variable structure and optimal control [57]. Though these control strategies exhibit a good performance, they are complex and difficult to implement [58]. The conventional proportional integral derivative (PID) controller exhibits good performance for linear system, and it is widely employed in the industry due to its simple structure and robustness in different operation condition. However, the accurate tuning of the parameters of PID become difficult because most of the industrial plant are highly complex and have some issues such as nonlinearities, time delay, and higher order [59].

Due to the complexity of most industrial plant and the limitation of PID controller, an unprecedented interest was diverted to the applications of the type-1 FLC. This is because it uses the expert knowledge and its control action is described by linguistic rules. Also, the FLC does not require the complete mathematical model of the system to be controlled and it can work properly with nonlinearities [7, 49, 51]. However, when there is uncertainty in the system the design of MF suffers with some difficulties as stated in the introduction. T2FLC can be used efficiently for nonlinear and complex system with uncertainty. The control performance of T1FLS can be improved by T2FLS because it has the advantage of FOU that can be used to improve the corresponding MF [60].

3.1 Structure of general fuzzy PID controller

For easy demonstration, the two input [error $e(t)$ and the error variation $\Delta e(t)$] and one output u direct action type of fuzzy PID controller (FPIDC) is used as shown in Fig. 4 (adapted from [61]). The number of input/output depends on the problem to be solved. In most real applications, the inputs are more than two. Detailed on fuzzy PID, fuzzy PD and fuzzy PI structures and their relationships are presented in [49, 62]. In addition, the structure of single input IT2FLC was discussed in literature [63–66], refer to Sect. 3.2 for details. The structure of FPIDC was formed from fuzzy PD controller with an integrator at the output. The output u is the control signal and is defined by Eq. 14 [61]. The scaling factors K_p, K_D, γ, δ are used to normalize the input/output of the FLC. The $e(t)$ and $\Delta e(t)$ are normalized by the scaling factor (K_p, K_D) to the common interval $(-1, 1)$ in which the MFs of the input are defined. After the normalization, $e(t)$ and $\Delta e(t)$ are converted to E and ΔE, respectively. γ and δ are used to map the output U onto the actual output domain u. In Fig. 4, there are two blocks that can be used to introduce the uncertainty in the system either in series with controller and plant or in the feedback or both. The structure in Fig. 4 is the same for T1FLC, IT2FLC and GT2FLC.

The design parameters of FLC can be categorized into two, namely tuning parameters and structural parameters. The tuning parameters include MF parameters and scaling factors and are to be adjusted offline at the design phase. Subsequently, the optimal parameters are fixed during the operation phase except in a case of continuous adaptation [49, 61]. The structural parameters include the inference mechanism, fuzzy rule, type of MF, fuzzy linguistic set and input/output variables to the fuzzy inference. The structural parameters are determined during the offline design phase [67, 68].

The major disadvantage of T2FLC is that the process of finding the appropriate values of parameters and structure become more difficult and more time consuming as the number of input and output of the system increases. This problem can be solved by using meta-heuristic optimization algorithms [69].

$$u = \delta U + \gamma \int U \mathrm{d}t \tag{14}$$

3.2 Single input interval type-2 fuzzy control (SIT2FLC) systems

As earlier mentioned in the introduction, there are experimental evidences illustrating significant improvements of T2FLS over T1FLS in terms accuracy. This is due to the extra degree of freedom provided by the FOU in the T2FS. However, the design and robust stability analysis of the IT2FLCs are still challenging problems owing to their relatively complex internal structure. Thus, SIT2FLC was proposed because of its simple structure. The essential feature of the SIT2FLC is the closed form output presentation which is defined in a two-dimensional domain. Moreover, the SIT2FLC structure has some preferred features of the PID controller such as easy design and simplicity [64]. The following literature describes the simple design and robust stability analysis for IT2FLC using SIT2FLC:

Kumbasar [63] pioneered the design methodology for single Input IT2FPID controller. The proposed simple structure provides an opportunity to design a T2FLC output in closed form formulation. This cannot be achieved with T2FPID controller structures with the Karnik–Mendel type reduction. The solution in closed form is derived based on the tuning parameters that are selected as the heights of lower MF of the antecedent IT2FSs. The author shows how the extra degrees of freedom of antecedent IT2FSs can be used to improve the control performance on both nonlinear and linear systems using simulations. Furthermore, the T2FLC structure has been implemented on experimental

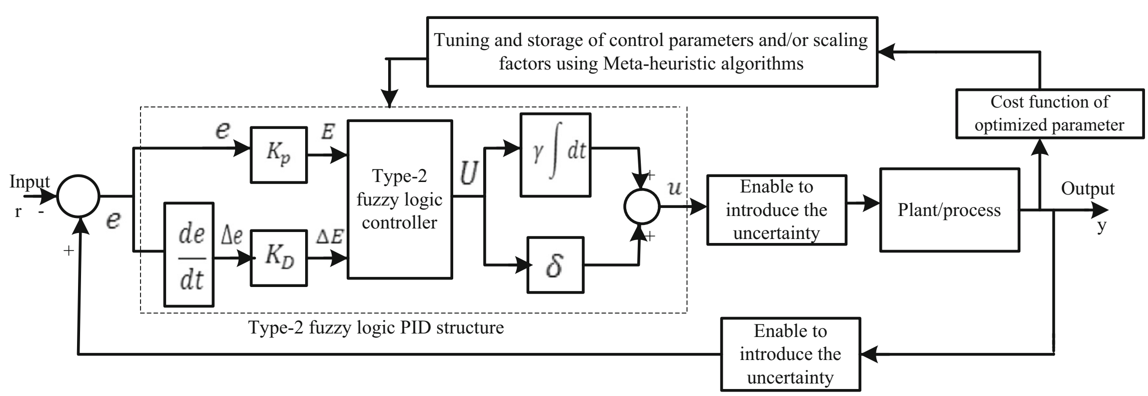

Fig. 4 Comprehensive block diagram of FLC (FLPIDC)

pH neutralization. The experimental and simulation results indicate that the proposed single input T2FLC has a superior performance and can cope with parameter uncertainties, nonlinear dynamics, disturbances and noise than the conventional PID controllers.

Kumbasar [65] study the robust stability of a PD type SIT2FLC structure by means of Lyapunov's direct method and Popov criterion approach. The type-2 fuzzy functional mapping (T2F-FM) was analyzed in a two-dimensional domain in view of the fact that a closed form formulation of the SIT2FLC output is possible. The author presents the mathematical derivations to confirm that T2F-FM is a symmetrical function and always sector bounded. As a result, the T2FS can be transformed into a perturbed Lur'e system in order to examine its robust stability. It has been confirmed that the stability of the proposed system is guaranteed based on Popov–Lyapunov method. A measure of robustness of the T2FLS was present to provide the bound of allowable nonlinearities and/or uncertainties of the control system. An illustrative example is provided to show the robust stability analysis of the proposed system. In [64], the functional mapping (FM) of a SIT2FLC was explicitly derived to present design techniques and examine its robustness. Comparative theoretical investigations are presented on the differences between the type-1 FM and IT2-FM to show the function of the FOU on the robust control system performance. It has been confirmed that the stability of the proposed system is guaranteed based on Popov–Lyapunov method. The author presents the analytical design techniques for SIT2FLCs to generate commonly used control curves by tuning the size of the FOUs only without requiring optimization method. It has been theoretically shown that the FOU provides the opportunity to the SIT2FLC to generate typically used nonlinear control curves and giving a certain level of robustness that cannot be accomplished by Type-1.

Kumbasar and Hagras [66] proposed the design of SIT2-FLCs based on an online tuning mechanism to improve their control performance. The structure of the PI type SIT2FLC and its IT2-FM with respect to the parameters of the FOU was presented. Based on this structural information, the authors presented a design strategy for SIT2FLCs which composed of three rules. This produces aggressive SIT2FLC (ASIT2FLC) and a smooth SIT2FLC (SSIT2-FLC) by only tuning a single parameter. The results obtained demonstrated that the SSIT2FLC has more robust control performance compared to ASIT2FLC. On the other hand, the disturbance rejection ability and the transient state of the SSIT2FLC could likely degrade compared to the ASIT2FLC. This problem was solved by tuning the size of FOU of the SIT2FLCs to give a trade-off between the acceptable transient and disturbance rejection performance of the ASIT2FLC structure and the robust control performance of the SSIT2FLC. Thus, a gradient-descent (GD)-based online tuning mechanism was introduced to improve both the transient state and disturbance rejection performances of the SIT2-FLCs at the same time preserving a certain level of the robustness against different conditions (disturbances and nonlinearities). The simulation results indicates that the GD-based SIT2FLC improve both the transient state and disturbance rejection performances compared to the robust self-tuning T1FLC and IT2FLC.

4 Method used for searching the publications

The papers considered in this work were found using the search engine of Scopus online system of Elsevier, where publications can be searched by author name or by subject. This database contains almost all the relevant and respected publications around the world. The complete publications on T2FLS optimization for application in intelligent control from the year 2012 to date were searched by using the following key words: type-2 fuzzy controller optimization using HO (several combinations of optimization methods were tested). The same search was done for GAs and PSO. Furthermore, the following optimization methods have also been searched: ACO, big bang–big crunch (BB–BC) optimization, BPA, bacterial foraging optimization, biogeography optimization, CO, simple tuning optimization and Tabu search optimization. Several other publications have been found for each optimization method. Also, other optimization methods were searched, but no publication was found in this area of the research. Similar criteria were used in [70].

5 Meta-heuristic optimization algorithms in optimizing a T2FLCs

A concise overview of the basic concepts of HO methods, GA, and PSO is presented in this section to give a brief description of how the algorithms operate. All the above mentioned algorithms can be used to solve the problem of designing the T2FLS. The critical issues in these optimization algorithms with respect to the design of FLS are: (1) encoding (representation) of fuzzy logic in the corresponding optimization paradigm, for example, the feature of FLS has to be encoded in form of chromosome for GA, firefly for FA, etc.; (2) determination of the boundaries of parameters to be optimized (solution space); (3) fitness function [28].

5.1 Hybrid optimization (HO)

HO method in T2FLS design refers to a combination of two or more meta-heuristic optimization algorithms for adjusting the parameters associated with T2FLS to speed up the optimization task by getting lower complexity in computation, faster convergence and global optimization [71]. In some cases, achieving an optimal solution to a problem using conventional optimization techniques that have high computational complexity is quite difficult [72]. Recently, the applications of HO techniques in automatic design of T2FLS for different complex applications have attracted interest from many researchers [73–75]. Taking the advantages of two or more optimization techniques, a HO which gives better convergent rates and solution quality can be found. More explanation about the automatic switching among the constituent optimizers in HO can be found in [76].

5.2 Genetic algorithms (GAs)

The basic foundation of GAs was proposed in 1975 by John Holland [77]. It is based on Darwin's ideas. Darwin's stated that in a competing environment, the stronger individuals are more likely to be the winners [78]. GA is a meta-heuristic search algorithm based on natural selection and genetic process [77, 78]. In GA, the potential solution to a problem is an individual which can be represented by a set of parameters. These parameters are just like a gene of a chromosome and can be represented by the string of values in binary form [77, 79]. The fitness value is used to test the degree of goodness of the chromosome for solving a problem that is directly related to the objective value. The operators employed in a simple GA include selection, crossover and mutation [77–81]. GAs is often regarded as function optimizers, and they have been applied in many optimization problems. In particular, the use of GAs for fuzzy systems design equips them with the adaptation and learning capabilities which brought about genetic fuzzy systems (GFSs) [79–81]. Different levels of complexity are covered by genetic learning processes according to the structural changes created by the algorithm [82], from optimization of parameters (simplest case) to learning the rule set of a rule-based system (highest level of complexity) [83]. The optimization of the parameter is the approach used to adapt a different variety of fuzzy system, as in genetic neuro-fuzzy systems or genetic fuzzy clustering [84]. It was reported in [81] that genetic fuzzy rule-based systems is the most well-known types of GFSs. It is essential to differentiate between learning and adaptation (tuning) problems in fuzzy systems. This is explained in Table 1 [85].

The flowchart for simple GA for optimization of T2FLS is shown in Fig. 5 (adapted from [28]). The GA was found to be good in the optimization of co-active neuro-fuzzy inference system [86].

5.3 Particle swarm optimization (PSO)

The PSO was introduced in by Eberhart and Kennedy [87]. It is an optimization algorithm based on social and population behavior, just like flocking of bird or fish schooling. The population in PSO is called swarm that can contain many particles. At each iteration t, the position P_t^i of the ith particle is updated using Eq. (15) [88]. The set S is updated to the next iteration using Eq. (16) [88].

$$V_{t+1}^i = V_t^i + c_1 r_1 \left(P_{\text{best}}^i - P_t^i\right) + c_2 r_2 \left(g_{\text{best}} - P_t^i\right) \tag{15}$$

$$P_{t+1}^i = P_t^i + V_{t+1}^i \tag{16}$$

where P_{best}^i is the best position attained for the individual particle and g_{best} is the best position attained for the particle among all the population. r_1 and r_2 are random numbers between 0 and 1, while c_1 and c_2 are position constants learning rate.

Modified PSO in the form of a constriction factor X, was introduced in [89, 90] and is given by Eqs. (17) and (18). In this, X controls the entire three components in the update velocity rule, in order to reduce the velocity as search progresses.

$$V_{t+1}^i = X\left[V_t^i + c_1 r_1 \left(P_{\text{best}}^i - P_t^i\right) + c_2 r_2 \left(g_{\text{best}} - P_t^i\right)\right] \tag{17}$$

$$X = \frac{2}{\left|2 - \beta - \sqrt{\beta^2 - 4\beta}\right|}, \quad \beta = c_1 + c_2 > 4. \tag{18}$$

The velocity is restricted within $[-V_{\max}, +V_{\max}]$. If the velocity deviates from this range, it has to be forced to be within the range [88, 89].

PSO algorithm can be implemented easily and demonstrates stable convergence when compared with other optimization algorithms [69, 91–93]. PSO has the following advantages over other conventional optimization method [69]: PSO is less vulnerable to being trapped in local minima because it is population-based search algorithm and exhibits implicit parallelism. PSO can also easily deal with non-differentiable and nonlinear objective functions. In addition, PSO is more robust and flexible than conventional methods, this is because it uses probabilistic transition rules rather than deterministic ones. Moreover, PSO has a unique feature of the suppleness to accomplish a sound balance between the local and global exploration of the search space. This enhances the overall search capabilities and overcomes the premature convergence problem, unlike GA and other heuristic algorithms. The quality of the solution in PSO is independent of the initial population, unlike the traditional methods.

The flowchart for simple PSO applied to a T2FLS is shown in Fig. 6 (adapted from [28]).

Table 1 Difference between learning and adaptation problem in fuzzy systems

Learning	Adaptation
Involve the process of automated fuzzy rule sets design that starts from scratch	Optimization of an existing fuzzy rule-based systems
Learning process perform a more elaborated search in the space of possible rule-base or entire knowledge base, irrespective of the predefined set of rules	Adaptation process assume a predefined rule-base and have the objective of searching a set of optimal parameters for the data-base, scaling function and MF

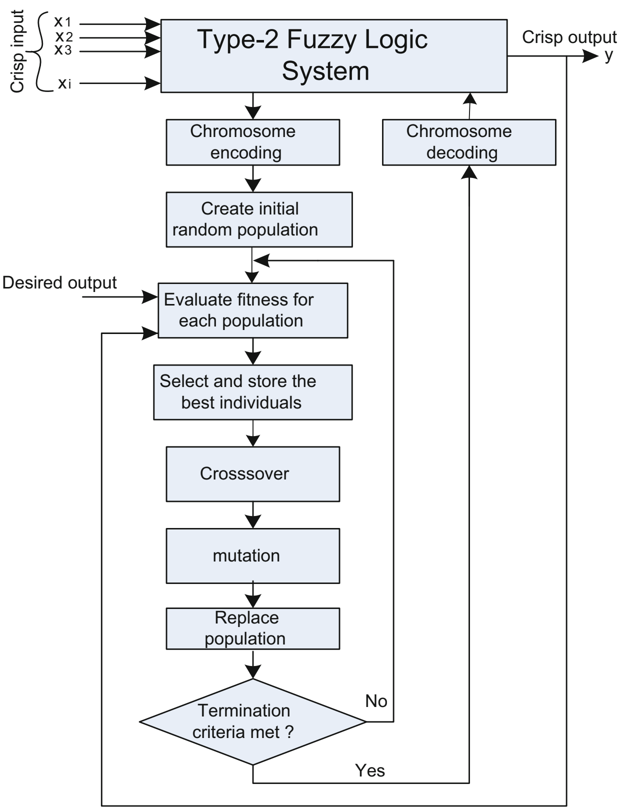

Fig. 5 Flowchart of simple GA for optimization of T2FLS

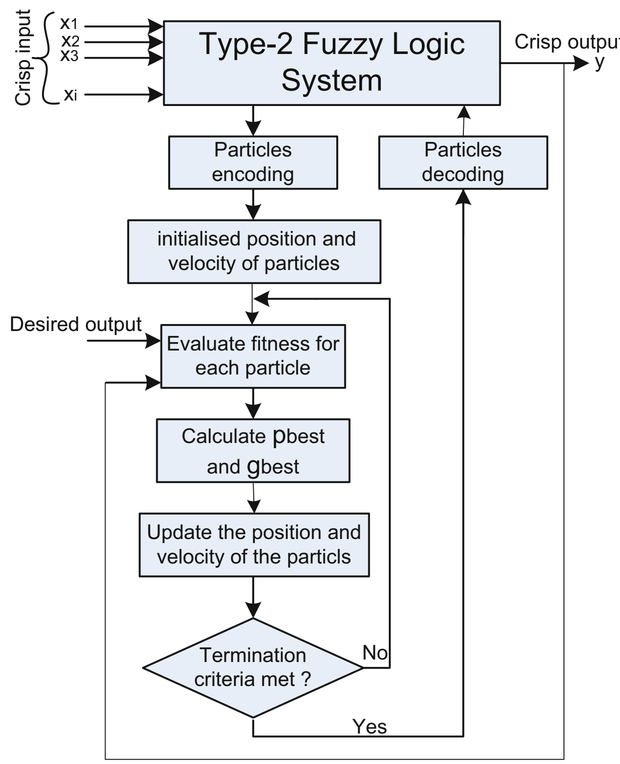

Fig. 6 Flowchart for simple PSO for optimization of T2FLS

5.4 Optimization of T2FLC by HO

There are many researches on the optimization of T2FLC using HO algorithms. Success was recorded in most of these researches. The review of these literatures is discussed in this section. The review in this section demonstrates the effectiveness of using the HO algorithms for automatic design of the T2FLC.

Martínez-Soto et al. [15, 56, 94] presented automatic design of FLC using Hybrid PSO-GA method for minimizing a steady state error of a plant's response. Three different plants were used as a benchmark, namely, stable system, unstable system and trajectory tracking control for autonomous mobile robot. Hybrid PSO-GA method was used to adjust the parameters of the FLC. To demonstrate the effectiveness and robustness of the hybrid PSO-GA based FLC, comparison was made between GA, PSO and Hybrid PSO-GA based FLC in different plant. The simulation results show that the IT2FLC and T1FLC obtained using hybrid PSO-GA was better than that of GA and PSO. In addition, the results obtained by IT2FLC are better than that of T1FLC in presence of disturbances. They used objective function that minimizes the steady state error using the average of the absolute error represented with the following equation:

$$\mathrm{EP_T} = \sum_{i=n} \frac{\mathrm{abs}(r_f(i) - c_r(i))}{2} \quad (19)$$

where r_f is the trajectory reference, c_r is the control response and $\mathrm{EP_T}$ is the average absolute error.

In Li et al. [95], the species-based hybrid electromagnetism-like mechanism (EM) and back propagation (BP) algorithm (SEMBP) were hybridize for the design of interval type-2 neural systems with asymmetric MF (AIT2FNS). The interval type-2 symmetric MF and the TSK-type consequents part are adopted for the implementation of the network structure in AIT2FNS. SEMBP was used to train AIT2FNS. The simulation results obtained by nonlinear tracking control and bath temperature control show the performance and effectiveness of the proposed method over the traditional PSO, GA, EM and SEM. They used the objective function defined as:

$$E(\cdot) = \frac{1}{2}\sum_{k}(y_{\mathrm{r}}(k) - y(k))^2 \tag{20}$$

where $y_{\mathrm{r}}(k)$ is the desired trajectory, $y(k)$ is the system output, and k is the discrete time index.

Fayek et al. [12] presented the systematic design and HO whereby PSO and GA optimize different parameters in the same system and real-time implementation of IT2FLC for control of position of DC servo motor. PSO and GA were used for the optimization of controller input/output gains, and MFs parameters, respectively. It was reported that the proposed controller outperformed the T1FLC and PI controller in terms of efficiency and effectiveness under noise and disturbances. The objective function used in both stages of the design process is a multiple-objective function defined as:

$$\mathrm{FE} = O_1 - 0.5O_2 \tag{21}$$

where $O_i = \mathrm{IAE} = \sum_{k=1}^{n}|e(k)|$, O_1 is the first objective function, O_2 is the second objective function, $e(k)$ is the error at kth point, and n is the number of points in the run.

Hsu and Juang [96] hybridized a species-differential-evolution and continues ant colony optimization (SDE-CACO) algorithms for improvement of IT2FLC performance. New species SDE mutation operation was introduced into a continuous ACO algorithm in order to improve its explorative ability. The clustering-based approach was used to generate all the IT2FLC rules online during the evaluating leaning process. SDE-CACO was used to optimize the free generated parameters of IT2FLC online. The proposed algorithm was applied for simultaneous wall-following control and speed control of mobile robot. Comparison was made between the continuous ACO, PSO and DE. Based on the real-world experiments and simulation results obtained, it was reported that the proposed algorithm showed higher efficiency and effectiveness in wall-following control and speed control over comparative algorithms. They introduced the new objective function that includes the following three principle factors for successful wall-following control strategy: (1) making smooth changes in steering angle; (2) moving at high speed; (3) maintaining a proper distance from the wall being followed. The overall control consists of two training stages. Each stage has different objective function C_1 and C_1 for stage one and stage two, respectively, as follows:

$$C_1 = f_1 + f_2 + f_3 \quad \text{and} \quad C_2 = f_1 + f_2 + f_3 + f_4 \tag{22}$$

where $f_1 = \frac{\sum_{t=1}^{T_{\mathrm{Total}}}|\mathrm{RD}(t)-1|}{T_{\mathrm{Total}}}$; $\mathrm{RD}(t) = \frac{S_4(t)}{d_{\mathrm{wall}}}$; $f_2 = \alpha_2 \times \frac{\sum_{t=1}^{T_{\mathrm{Total}}}|\phi(t)|}{T_{\mathrm{Total}}}$; $f_3 = \alpha_3 \times (T_{\mathrm{Total}} - T_{\mathrm{stop}})$; $f_4 = \alpha_4 \times \frac{1}{\sum_{t=1}^{T_{\mathrm{Total}}} d(t)/T_{\mathrm{Total}}}$. f_1 is the average of the relative RD values occurring over all T_{Total} time steps. f_2 describes the included angle φ between the robot front direction and the wall. f_3 describes the difference between T_{Total} and the time step number T_{Stop} when a collision occurs. f_4 is defined as the inverse of the average moving speed. α_2, α_3 and α_4 are weighting coefficients.

A summary of presented publications where HO has been used to optimize the T2FLC is presented in Table 2.

5.5 Optimization of T2FLC by GAs

Several studies have been done on optimization of T2FLC using different variant of GA. Success was reported in most of these works in different area of control applications. Thus, this section presents the state-of-the-art review.

Sun et al. [13] proposed RNA genetic algorithm (RNA-GA) for the optimization of MFs parameters associated with T2FLC and T1FLC. Five nonlinear functions constraints were used to test the searching capability of the RNA-GA. The performance of the optimized T2FLC and T1FLC using RNA-GA and GA were tested on control of the double inverted pendulum system under unexpected disturbances. Integral of time weighted absolute value of error (ITAE) was used as objective function. Based on the experimental results obtained, it was found that the optimized T2FLC demonstrates superiority in the elimination of obstinate vibrations and oscillations over the optimized T1FLC. In addition, comparative simulations show that the RNA-GA optimized T2FLC better than the comparative algorithms. Lu's paper [19] proposes an IT2FLC with GA-based type reduction algorithm for reduction of IT2FS as well as to obtain the optimal defuzzified output from type reduced set. The root-mean-square error (RMSE) was used as an objective function. The proposed type reduction was executed offline which reduced the computational cost significantly and facilitate the design of IT2FLC operation in real time. The proposed controller was applied on truck backing control problem. The proposed IT2FLC outperformed the conventional IT2FLC in terms of robustness, computational cost and speed.

Cervantes and Castillo [97] proposed a novel method for complex control by combining several FLC. The proposed

Table 2 Optimization of T2FLC using HO

References	Hybrid optimization	Algorithm/s compared with	Controller/s compared with	Results
Martínez-Soto et al. [15, 56, 94]	PSO + GA	GA and PSO	T1FLC	PSO + GA better than GA and PSO. T2FLC outperform T1FLC
Li et al. [95]	EM + BP	PSO, GA and EM	IT2FNS	EM + BP better than PSO and GA. AIT2FNS better than IT2FNS
Fayek et al. [12]	PSO + GA	Not compared	T1FLC and PI	T2FLC better than T1FLC and PI
Hsu and Juan [96]	SDE + CACO	ACO, PSO and DE	Not compared	SDE + CACO better than ACO, PSO and DE

method is particularly useful for multivariable control system. The method has two levels of hierarchical architecture (individual FLC and a superior control to adjust the global result). Flight controls that require several individual controller was used to test the behavior of the proposed method. GA was used to adjust the parameter of the T2FLC. The fitness function that quantifies the errors of each controller was used as follows:

$$f(Y) = \frac{\left(\sum_{i=1}^{n} \frac{|y^i_{\text{ref}_1}|}{n} - \frac{|y^i_{\text{fs}_1}|}{n} + \sum_{i=1}^{n} \frac{|y^i_{\text{ref}_2}|}{n} - \frac{|y^i_{\text{fs}_2}|}{n} + \sum_{i=1}^{n} \frac{|y^i_{\text{ref}_m}|}{n} - \frac{|y^i_{\text{fs}_m}|}{n}\right)}{m} \tag{23}$$

where y_{ref} is the reference, y_{fs} is the output of the controller and n is the number of points of the dynamic response used in the comparison, m stands for the number of the individual controllers used. Based on the simulation result obtained, it was concluded that the proposed method use T2FLC and it decreases the control error and improve the overall behavior of the plant when compared with the T1FLC.

Maldonado et al. [98] authors optimized the average approximation of an interval type-2 fuzzy logic controller (AT2FLC) using multi-objective GA for hardware applications such as speed control of DC motor in FPGAs. The researchers considered the combination of triangular and trapezoidal T2-MFs of an AT2-FLC such that the GA needs to optimize some parameters (adaptable) of the T2-MFs in order to have less execution time. The composite objective function that compromise between the minimum overshoot, undershoot and steady state error was used as shown in Eq. (24). The optimized AT2FLC was compared with the optimized T1-FLC as well as the PID controller tuned by Ziegler–Nichols method. The real-world experiment's results show that AT2FLC outperformed the T1FLC and PID controller by observing the lower error in presence of uncertainty.

The composite objective function is as follows:

$$U = \sum_{i=1} \omega_i f_i(x); \tag{24}$$

where ω is the positive value.

Castillo and Cervantes [99] used GA in the design of a T1FLC and T2FLS for airplane longitudinal control. They used three inputs (stick, rate of elevation and angle of attack) to the controller. GA was applied for the optimization of MFs of the fuzzy systems. The fitness function used is based on the average error between the output of PID and fuzzy controllers as shown in Eq. (25). Based on the simulation results obtained, it was concluded that at higher levels of disturbances in the plant, the GA-based T2FLC outperform the T1FLC as well as the PID controller.

$$\text{Error} = \sum_{i=1}^{n} \frac{\left|y^i_{\text{PID}} - y^i_{\text{fuzzy}}\right|}{n} \tag{25}$$

where y_{PID} is the output of the PID controller and y_{fuzzy} is the output of the fuzzy controller.

Melendez et al. [100] proposed the GA for the optimization of an interval type-2 fuzzy reactive controller for autonomous mobile robot. The hierarchical genetic algorithm (HGA) was used for the optimization of fuzzy MF, fuzzy rules and mobile robot power usage. To overcome the problem of loop trajectory, the new module was added to the system which consists of a monolithic neural network that is used for detecting patterns of loop on the robot trajectory. Based on the simulation results obtained, it was concluded that HGA shows its effectiveness on multi-objective task and the proposed type-2 neuro-fuzzy allowed the HGA to optimize the forward movement of the robot through the maze and refrain from any type of collision with obstacles.

Cervantes and Castillo's paper [101] presented statistical comparisons of T1FLC and T2FLC for control of water level in three tanks. GA was used for optimization of MFs for both controllers. MSE was used as an objective function. Three different paradigms, namely, empirical T1FLC, GA-based T1FLC and GA-based T2FLC were used for comparisons. It was concluded that based on the simulation results obtained, the GA-based T2FLC shows better performance compared to empirical T1FLC and GA-based T1FLC. In the work of Bi et al. [102], a GA-based T2FLC for single intersection signal control was proposed. GA was used to optimize the MFs parameters of T2FLC. The comparison results (simulation) show the superiority of GA-based T2FLC over T1FLC as well as fixed time control in terms of reduction of vehicular delay and queue length at the traffic intersection. The objective function used is as follows:

$$\mathrm{OF} = \sum_{l=1}^{n} \frac{D_{\mathrm{Rn}}^{l} + D_{\mathrm{GN}}^{l}}{\mathrm{qg}^{l} + \mathrm{qr}^{l} + \mathrm{lqg}^{l-1} + \mathrm{lqr}^{l-1}} \tag{26}$$

where D_{Rn}^{l} and D_{GN}^{l} are the total delay time of the vehicles in red phase and total wait time at green phase of l cycle, respectively. qg^{l} and qr^{l} are the number of arrival vehicles at green phase and red phase. lqg^{l} and lqr^{l-1} are the staying number of vehicles at different phase in the previous cycle.

Shill et al. [103] applied a real-coded quantum GA for simultaneous optimization of T2FS and rule sets for control of robot manipulators with unstructured dynamical uncertainties. The main aim of their study is to make the design of IT2FLC automatic. It was concluded that based on the real-world experiment and simulation, quantum GA-based T2FLC resisted noisy unstructured environment and succeeded in having higher control performance better than the quantum GA-based T1FLC, traditional T1FLC, and neural coded FLCs.

Moldonado and Castillo [104] described the automatic design of an average type-2 fuzzy logic controller (AT2-FLC) for DC motor speed control. The GA and PSO were employed for optimization of parameters of AT2-FLC. The objective function of PSO and GA considers three characteristics that make them to be multi-objective type as follows: minimum overshoot: if $y_{(t)} > r_{(t)} \rightarrow o_1 = \min(y_{(t)}) - r_{(t)}$; minimum undershoot: $o_2 = |\min(y_{(t)}) - r_{(t)}|$; minimum output steady state error: $\mathrm{SSE} = \sum_{t=201}^{1000} y_{(t)} - r_{(t)}$ where $y_{(t)}$ is the output of the system and $r_{(t)}$ is the reference signal. Simulation was carried out in FPGA using the xilinx system generator. Satisfactory results were obtained for both GA and PSO. However, the result obtained by GA was better than the results obtained by PSO. In another work, Moldonado and Castillo [105] used GA-based T2FLC for velocity regulation in DC motor. Xilinx system generator of Xilinx ISE, and MATLAB–Simulink for synthesizing the T2FLC in a very high description language (VHDL) code for a field programmable gate array (FPGA) was used. GA was used to optimize trapezoidal and triangular MFs of T2FLC and T1FLC for hardware application such as FPGA. The same objective function as the one presented in [104] was used. The GA-based T2FLC, GA-based T1FLC and PID controller tuned by Ziegler–Nichols method was compared. It was reported that GA-based T2FLC outperformed GA-based T1FLC as well as PID controller.

Li et al. [106] applied a single input rule modules (SIRM)-connected T2FLC scheme for baking up control of the truck-trailer system. The two most important tasks for designing such SIRM-connected T2FLC are: design a suitable SIRM for each input item and determine the parameters of SIRM-connected T2FLC such as important degree of input items, scaling factors and MF of T2FLC in each SIRM. GA was used to optimize these parameters. The control objective is to force the vertical position y, trailer angle β, and relative angle α to be zero. The following fitness function was used:

$$J = \sum_{s=1}^{S} \left(\gamma_{\alpha} \mathrm{IAE}_s(\alpha) + \gamma_{\beta} \mathrm{IAE}_s(\beta) + \gamma_{y} \mathrm{IAE}_s(y) \right) \tag{27}$$

where S is the number of initial conditions used. γ_{α}, γ_{β} and γ_{y} are scaling factors. The comparison between T2FLC and T1FLC was made. The simulation results show that the proposed scheme with optimized T2FLC outperformed the scheme with optimized T1FLC and gave less time backing up the truck-trailer system from initial state to expected state. Ghaemi et al. [107] presented the hybrid combination of adaptive type-2 fuzzy logic control and sliding mode control (SMC) called adaptive interval type-2 fuzzy proportional integral sliding mode controller (AIT2FPISMC) for the design of a robust control system with high level of uncertainties and nonlinearity. The T2FLS was used to approximate unknown nonlinear terms, PI was used to eliminate chattering effect in SMC, and SMC was used to address the issue of noise and disturbances. GA was employed to tune the parameters of AIT2FPISMC. Mean square error (MSE) for the closed loop control was considered as the fitness function. It was found that based on the simulation results, the proposed controller has good performance and was improved with GA. A summary of review through which GAs have been used to optimize the T2FLC is presented in Table 3.

5.6 Optimization of T2FLC by PSO

Many publications on optimization of T2FLC using different kinds of PSO exist in the literature. Thus, we review the literature that used PSO for the optimization of automatic design of T2FLC.

Hamza et al. [108] present the design of an optimal Interval type 2 fuzzy proportional derivative controller (IT2F-PDC) in cascade form, for rotary inverted pendulum (RIP) system. The parameters associated with the IT2F-

PDC are optimized using GA and PSO. The performance characteristics considered for the controllers are steady state error E_{ss}, settling time t_s, rise time t_r, maximum overshoot M_p, and control energy E_u as follows:

$$\cos t = \frac{1}{2}\left(M_p + E_{ss} + E_u\right) - \frac{e^{-\gamma}}{2}\left(t_r - t_s + E_{ss} + M_p\right) \tag{28}$$

where γ is the weighing factor. Experimental and simulation results indicated that the effectiveness and robustness of the proposed GA-based and PSO-based controllers on the RIP with respect to load disturbances, parameter variation, and noise effects have been improved over energy based controller. However, the comparative results for simulation and experiment based on cascade IT2F-PDC indicate that GA-based IT2F-PDC has lower steady state error, while PSO-based IT2F-PDC has lower overshoot, settling time, and control energy, although both have almost the same rise time.

Shahsadeghi et al. [92] presented an optimal type-2 fuzzy sliding mode (OT2FSM) controller for control of general chaotic systems. The random inertia weigh PSO (RNW-PSO) was used to adjust the parameters of the controller including the input and output MFs coefficients of type-2 fuzzy. The inertia weight enhances the efficiency of the PSO by equilibrating the local exploitation and the global exploration capabilities of the swarm. The MSE for the closed loop control was considered as the fitness function. The simulation results obtained by the proposed controller were compared with that of optimal type-2 FPIDC and optimal H-infinity adaptive PID controller. It was found that the proposed controller perform better than the comparatives controllers. Niknam et al. [109] presented an optimal type-2 sliding mode controller for class of nonlinear uncertain systems under external disturbances. RNW-PSO was used to adjust the parameters of the controller as well as the input and output MFs coefficients of type-2 fuzzy. The mean root squared error (MRSE) was used as an objective function. Inverted pendulum system was used as case study. Based on the simulation results, it was concluded that the proposed controller is free of chattering effect, robust and obtained the desired performance. Khooban et al. [110] proposed the design of an optimal type-2 FPIDC for air supply pressure of air-conditioning, heating and ventilation systems. The coefficient of PID and the input and output MFs parameters of IT2FLC were simultaneously optimized by RNW-PSO. MSE was considered as the fitness function. The simulation result shows that the proposed controller outperformed the PID, ANF and STFPIC controllers under the presence of the uncertainties in the parameters of the model.

Allawi and Abdallah [111] proposed an IT2FLC for multiple mobile robots. The researchers considered the control of robot cooperation, target reaching task and avoidance of collision during navigation. The parameter of the proposed controller was adjusted using PSO. Fitness

Table 3 Optimization of T2FLC using GAs

References	Algorithm(s) compared with	Controller(s) compared with	Results
Sun et al. [13]	RNA-GA	T1FLC	RNA-GA better than GA. T2FLC better than T1FLC
Lu [19]	Not compared	Conventional IT2FLC	Proposed IT2FLC outperformed conventional IT2FLC
Cervantes and Castillo [97]	Not compared	T1FLC	T2FLC better than T1FLC
Maldonado et al. [98]	Not compared	T1FLC and PID	AT2FLC better than T1FLC and PID
Castillo and Cervantes [99]	Not compared	T1FLC and PID	T2FLC better than T1FLC and PID
Melendez et al. [100]	Not compared	Not compared	Not compared
Cervantes et al. [101]	Not compared	T1FLC	T2FLC better than T1FLC
Bi et al. [102]	Not compared	T1FLC and fixed time control	T2FLC better than T1FLC and fixed time control
Shill et al. [103]	No	No, T1FLC and neural coded FLS	T2FLC better than T1FLC and neural coded FLS
Moldonado and Castillo [104]	PSO	NO	GA better than PSO
Moldonado and Castillo [105]	Not compared	T1FLC	T2FLC better than T1FLC
Li et al. [106]	Not compared	T1FLC	T2FLC better than T1FLC
Ghaemi et al. [107]	Not compared	Not compared	Not compared

function which constitutes the number of robots and collision time was used as follows:

$$\text{fitness}_i = \frac{1}{2}\sum_{j=1, j\neq i}^{n}\left(\frac{k}{\text{tc}_{ij}}\right)^2 \quad (29)$$

where n is the number of robots, tc_{ij} is the collision time between robot i and robot j and k is the constant. Hybrid reciprocal velocity obstacles were used for avoidance of collision. They used two real E-peck mobile robots for experimental testing. It was concluded that the robot navigation efficiency was increased for optimized IT2FLC in both simulation and experimental results compared with IT2FLC. Maldonado et al. [93] presented the optimization of MFs of average approximation of an interval type-2 fuzzy logic controller (AT2FLC) using PSO. The same objective function as presented in [104] was used. To minimize the running time, the fuzzy rules were not modified and only certain point of MFs were considered. The proposed method was applied for regulation of speed of DC motor, and it is coded in VHDL for a FPGA Xilinx Spartan 3A. The experimental results obtained by PSO were compared with the one obtained by GA which indicated that, PSO has faster running time than GA. Baklouti and Alimi [14] propose a new adaptive learning procedure of IT2FLC for design of robot navigation planning task. Real-time PSO technique was used to instantaneously optimize the MFs of the IT2FLC. The fitness function that minimize the angular velocity smoothness index was used. The “iRobot create” was used as benchmark. It was concluded that based on the real-world experimental results found, the proposed method gives a free collision trajectory and smooth path for navigation of the robot. Panda et al. [112] presented a PSO-based IT2FLC for the design of power system stabilizer for damping oscillations in power transmission line. The proposed controller was tested on the single machine infinite bus and multiple machine infinite bus. PSO algorithm was used to optimize the MFs of the IT2FLC. The objective function of PSO that minimize four characteristics (settling time, peak time, undershoot and overshoot) was considered. The proposed controller was compared with the optimal T1FLC and optimal lead-lag controller. Based on the simulation results obtained, it was reported that the proposed controller was found to be more robust with respect to the different disturbances and system parameters variations. A summary of the publications in which PSO was applied for the optimization of the T2FLC is presented in Table 4.

5.7 Other meta-heuristic optimization algorithms in the optimization of T2FLC

Several works on optimization of T2FLC using different types of optimization algorithms have been reported in literature. Success was recorded in most of these researches in different areas of applications. The review of the representative of these types of research was presented in this section to demonstrate the effectiveness of using the corresponding optimization method for automatic design of T2FLC.

Castillo et al. [113] presented an ACO and PSO method for adjusting the MFs parameters of an IT2FLC for design of an optimal intelligent controller for trajectory tracking control of autonomous mobile robot. The MSE for the closed loop control was considered as the fitness function. The statistical analysis shows that the ACO outperformed PSO and GA in this particular control and IT2FLC outperformed T1FLC. Yesil [114] propose a BB–BC optimization design strategy of interval type-2 fuzzy PID (IT2FPID) controllers for load frequency control in power systems. The BB–BC optimization was used to adjust the FOU, MFs and the scaling factor of IT2FPID controllers. The integral of time and absolute error (ITAE) was used as objective function. Four area interconnected power system was used as a benchmark. Based on the simulation results, the comparisons were made between the proposed method, T1FLPID and conventional PID, while all the controllers were optimized using BB–BC. The IT2FPID controllers operate 51.5 and 76.5 % better than the T1FPID and PID, respectively. A novel application of BB–BC for optimization of cascade structure of IT2FPID and its antecedent MF parameters was presented by Kumbasar and Hagras [22]. The proposed controller was applied for the path tracking control of PIONEER3-DX mobile robot. Several experiments in both simulation and real world were performed. IAE was used as a performance measure for both inner and outer loop. The proposed controller was compared with self-tuning T1FPID structure as well as PID and T1FPID counterpart which were optimized using BB–BC. The results obtained illustrate that the IT2FPID structure enhanced the control performance significantly in the presence of disturbances and uncertainties when compared with self-tuning, BB–BC-based PID and BB–BC-based T1FPID structures. In addition, they also show that the reason for superior control performance of IT2FPID controller is not just because it has extra parameters, but rather is in the way it is dealing with noise and uncertainty present in real world compared with self-tuning T1FPIDC. In addition, BB–BC gives high-quality solution with less computational time when compared with PSO.

El-Nagar and El-Bardini [26] proposed the IT2FNN controller for hardware-in-the-loop simulation to simulate the control of a multivariable anesthesia system. The antecedent part consists of IT2F linguistic process, while the consequent part consists of interval neural network. BPA was used for online training of parameters of

Table 4 Optimization of T2FLC using PSO

References	Algorithm(s) comparison with	Controller(s) compared with	Results
Hamza et al. [108]	GA	Energy based controller	T2FLC better than energy based controller, PSO better than GA
Shahsadeghi et al. [92]	Not compared	PID	T2FLC better than PID
Niknam et al. [109]	Not compared	Not compared	Not compared
Khooban et al. [110]	Not compared	PID, ANF and STFPIC	T2FLC better than PID, ANF and STFPIC
Allawi and Abdallah [111]	Not compared	Unoptimized T2FLC	Optimal T2FLC better than unoptimized T2FLC
Maldonado et al. [93]	GA	Not compared	PSO better than GA
Baklouti and Alimi [14]	Not compared	T1FLC	T2FLC better than T1FLC
Panda et al. [112]	Not compared	T1FLC and lead lag controller	T2FLC better than T1FLC and lead lag controller

IT2FNN. The performance criteria used are ISE, ITAE and RMSE. The experimental result obtained by the proposed controller outperformed the one obtained by the adaptive IT2FLC and T1FNN controller under uncertainties in terms of settling time and overshoot. Kiani et al.'s paper [23] presented the optimal design of IT2FLC for automatic voltage regulator system. Bacterial foraging optimization algorithm (BFOA) was used for tuning the MFs of IT2FLC. They consider the following fitness function in the simulation:

$$\min_k J(k) = G_e \int_0^T e^2(t)\mathrm{d}t + G_u \int_0^T u^2(t)\mathrm{d}t + G_\mathrm{M} M_\mathrm{p} \tag{30}$$

where $e(t)$ is the error, $u(t)$ is the control signal, T is the running time, M_p is the overshoot, and G_e, G_u, and G_M are the weighted constants. The simulation results obtained indicated that the BFOA outperform the extended discrete action reinforcement learning automata (EDARLA) under noise. Sayed et al. [24] presented a modified biogeography-based optimization algorithm (MBBO) for design of an IT2FLC for the improvement of the performance of Egyptian second testing nuclear research reactor. MBBO was employed in designing the IT2FLC in order to get optimal parameters of MFs of the controller. ISE was used as an objective function. In the simulation, IT2FLC obtained the best performance and ISE index compared to PD controller. In another work by Sayed et al. [115], an optimal design of IT2FLC for performance improvement of the plant control systems was presented. The MBBO and PSO were used for tuning the MFs parameters of IT2FLC. ISE was used as an OF. The proposed controller was tested on two plants with different complexity (stable and unstable). The simulation results obtained by MBBO are better than the PSO based on running time and overshoot. Melin et al. [116] presented an optimal design of FLC for tracking problem of the dynamic model of a unicycle mobile robot under perturbed torques. CO was used for searching the optimal parameters of IT2FLC and T1FLC. ISE was used as an objective function. Both experimental and simulation results show that the CO outperform the GA, PSO and ACO. Also CO-based IT2FLC outperformed the CO-based T1FLC. Astudillo et al. [25, 117] applied the chemical reaction optimization (CRO) method for optimal design of IT2FLC for tracking control of unicycle autonomous mobile robot. CRO was used to search for the gain constants and MFs parameters involved in the tracking controller. ISE was used as an OF. Both experimental and simulation results show that the best error obtained by CRO was similar with the one obtained with GA with less running time.

Li et al. [118] proposed IT2FLS-based data-driven strategy for modeling and optimization of thermal comfort words and energy consumption. The multi-objective optimization was used to optimize the temperature range through energy consumption and balancing the thermal comfort including MFs of the FLS. The results found in the simulation indicated that the specific room showed the robustness and effectiveness of the proposed method. Cortes-Rios et al. [119] propose the parallel model implementation of simple tuning algorithm (STA) for IT2FLC. The effect of AND/OR operators combinations on fuzzy rules, new integral criteria parameters, and mechanism to calculate the feedback gain are included to improve the algorithm applicability and performance. ISE, IAE, ITSE and ITAE were considering as performance criteria. The STA-based IT2FLC was tested using Hagglund and Astrom benchmark systems, and their performance was compared with that of PID controllers. Good performance of the proposed method was achieved compared to PID controller based on simulation and experimental results.

Liu et al. [29] presented an optimal solution to a solid transportation problem (STP) with Tabu search algorithm (TSA)-based type-2 fuzzy variables. ISE was used as an OF. Three types of new defuzzification criteria for type-2 fuzzy variables were proposed such as expected value, optimistic value and pessimistic value. The multi fold fuzzy STP is formulated as the chance-constrained programming model with minimum cost of expected transportation. The application and effectiveness of the proposed method was illustrated using numerical experiments, and it was found to advance performance over state-of-the-art methods.

A summary of presented publications in which other optimization methods that have been used to optimize the T2FLC is presented in Table 5.

6 Future trend and general overview of the research area

The general summary of the area of this research, i.e., the applications of meta-heuristic optimization algorithms in the design of T2FLCs is presented in this section. Based on the review of this area, the possible future trends that can be visualized were discussed. The complex problems of different kind of control (robotic, power systems, motor systems, mechanical systems etc.) with high level of noise and/or uncertainties can be handled properly by type-2 fuzzy systems. In recent years, the applications of type-2 fuzzy systems in intelligent control have become a common practice. It is widely known that the FLS design is not a simple task, particularly the design of T2FLS (IT2FLS and GT2FLS). This is due to the fact that IT2FLS has more design parameters than the T1FLS and the GT2FLS has more design parameters than IT2FLS. GAs, PSO, ACO have been used in automatic design of type-1 as well as type-2 systems that made it to become a standard practice [11]. Now the trend has been extended to the use of the combination of more than one optimization algorithms in solving a problem. This method is referred to as HO.

The summary of total publications per year from 2012 to date in the area of T2FLC optimization is shown in Fig. 7a. It can be noted that the total number of publications for 2012 to date is increasing per year (although, at this moment the information of 2015 is not yet complete). It can be stated that the trend in the area of design of T2FLC using optimization method is increasing yearly and this trend is expected to continue in the future because T2FLSs have been used more frequently in this area of research. Based on this review, it is noteworthy to state that, to the best of our knowledge, most of the applications in the area of T2FLS use the IT2FLS and only few applications considered the use of GT2FLS like [27, 33, 34]. For example, Kumbasar and Hagras [35] presented the novel zSlices-based general type-2 fuzzy proportional integral controller (zGT2FPIC). The propose zGT2FPIC combine the advantages of interval type-2 fuzzy PI controller (IT2FPIC) (i.e., robust control performance) and type-1 fuzzy PI controller (T1FPIC) (i.e., acceptable transient and disturbance rejection performance). The secondary MF of the antecedent GT2FS are adjusted in an online manner. The effect of the secondary MF on the closed system control performance improvement was examined. PIONEER 3-DX mobile robot was used as a benchmark. It was concluded that based on the simulation and real-world experiment results obtained, self-tuning zGT2FPIC is more robust to noise, disturbances, and uncertainties when compared with the IT2FPIC and T1FPIC. Sanchez et al. [33] compares the performance of GT2FLC, IT2FLC and T1FLC for control of mobile robot. It was observed that based on the simulation results and in the presence of external noise (band-limited white noise, pulse noise and uniform random number noise), GT2FLC outperforms their IT2FLC and T1FLC counterparts. ITAE, ITSE, IAE and ISE were used as the performance indexes. The low application of GT2GLS in literature is due to the difficulty in processing and managing the GT2FS, but with the invaluable work presented in [10, 40–47], it is expected that GT2FLS will become more standard in the area of T2FLS, which will eventually need more powerful optimization techniques. It is worth mentioning that based on this review, only few papers used the optimization based design to employ T2FLC in real-world environment. The following literature implements the IT2FLC in real-world environment [12–14, 22, 25, 26, 96, 98, 103, 108, 111, 116, 119], while only [35] implements the GT2FLC in real-world environment.

The distribution of the published papers in optimization based design of T2FLC according to the different optimization methods mentioned previously is shown in Fig. 7b. It can be noted that, to this moment GA and PSO are in the front line in optimization of T2FLS. However, the application of HO in T2FLC is increasing. This is due to the fact that recently in more complex works, the HO is able to outperform (in terms of less running time, less computation and better in multi-objective task) the traditional optimization methods (such as GA, PSO, FA) in different applications.

It is hard to declare one of these techniques as the best for optimization of T2FLS at the moment. This is because all of the reviewed methods have a record as successful methods of optimization of T2FLS in some applications. Although the HO method has outperformed some conventional optimization methods (such as genetic algorithms, PSO and firefly algorithm) in different applications, it cannot be declared as the best since it is not compared with all of the methods. In any case, the need of application

Table 5 Optimization of type-2 fuzzy systems using other optimization algorithms

References	Algorithm proposed	Algorithm(s) compared with	Controller(s) compared with	Results
Castillo et al. [113]	Ant colony optimization	PSO and GA	T1FLC	T2FLC better than T1FLC and ACO better than PSO and GA
Yesil [114]	Big bang–big crunch	Not compared	T1FLC and PID	T2FLC better than T1FLC and PID
Kumbasar and Hagras [22]	Big bang–big crunch	PSO	Self-tuning T1FPIDC, BB–BC-based T1FPIDC and PID	T2FPID better than Self-tuning T1FPIDC, BB–BC-based T1FPIDC and PID
El-Nagar et al. [26]	Back propagation algorithm	Not compared	Adaptive IT2FLC and T1FLC	T2FLC better than adaptive IT2FLC and T1FLC
Kiani et al. [23]	Bacterial foraging optimization	EDARLA	Not compared	BFO better than EDARLA
Sayed et al. [24]	Biogeography optimization	Not compared	PD	T2FLC better than PD
Sayed et al. [115]	Biogeography optimization	PSO	Not compared	MBBO better than PSO
Melin et al. [116]	Chemical optimization	GA, PSO and ACO	T1FLC	T2FLC better than T1FLC and CO better than GA, PSO and ACO
Astudillo et al. [25, 117]	Chemical optimization	GA	Not compared	CO better than GA
Li et al. [118]	Multi-objective optimization	Not compared	Not compared	Not compared
Cortes-Rios et al. [119]	Simple tuning optimization	Not compared	PID	T2FLC better than PID
Liu et al. [29]	Tabu search optimization	Not compared	Not compared	Not compared

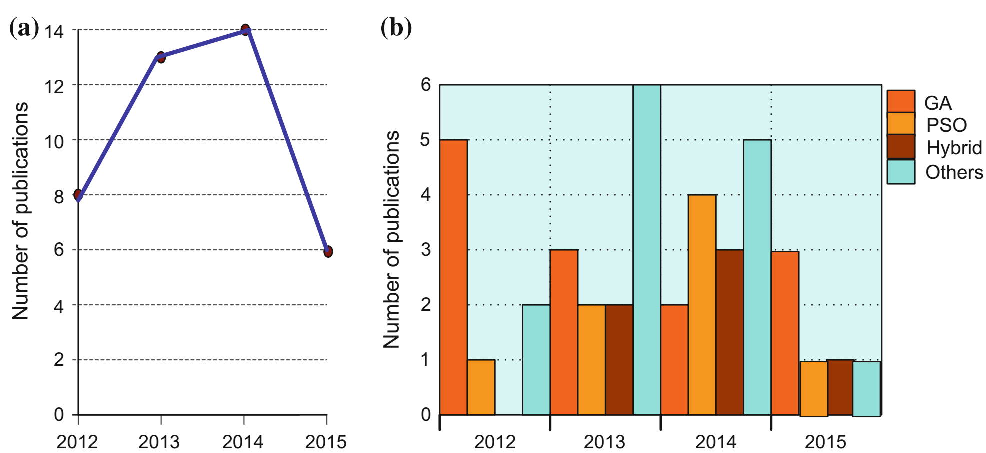

Fig. 7 **a** Total number of publications per year, **b** distribution of publications per type of optimization method and year

of optimization method in design of type-2 fuzzy systems was justified. This is due to the complexity in the design. Furthermore, it is advisable not to use gradient-based algorithms in optimizing T2FLS because the computations for MFs parameters will become much more complicated [10] Therefore, researchers in this area should use the derivative free algorithms such as QPSO, SA, PSO, GA ACO etc. or their combination to form hybrid. We suggest the use recurrent neural networks with hybrid meta-heuristic cuckoo search [120] for the optimization of T2FLS.

To the best of our knowledge, there are many types of optimization techniques that have not yet been applied to the optimization of T2FLC. These techniques include grid search, genetic programming, harmony computing, membrane computing, cuckoo search optimization, bat algorithm, bee colony optimization, simulated annealing, firefly algorithm (although some of these optimization methods have been applied to the design of type-2 fuzzy, common applications are for time series prediction, classification, clustering and pattern recognition as in literatures [27, 121]). Also, we recommend the optimization of T2FLC using flower pollination algorithm in view of the fact that it shows performance improvement over many meta-heuristic algorithms [122]. It is expected that these optimization methods as well as those that will be proposed in the future will be applied in further studies in order to determine their effectiveness in the area designing of T2FLS.

7 Conclusions

Evaluating the best values of parameters and structure in design of T2FLSs is a difficult task. Meta-heuristic optimization algorithms have been used in order to solve this problem. Recently, the use of HO is attracting tremendous attention in this area of research. A review on optimizing the design of T2FLCs was presented in this paper. This review is to justify the need of optimization of parameters associated with type-2 fuzzy systems for different applications. The application of HO, genetic algorithms and PSO is considered as the three algorithms that assist in optimal design of type-2 fuzzy controllers. To date, the most frequently used optimization methods in this area of research are genetic algorithms, particle swarm optimization and HO. The purpose of this review paper is to expose various optimization techniques for type-2 fuzzy controller to the researchers and use this as a guide for further advancement of research and explore other techniques that have received little or no attention.

Acknowledgments The authors would like to acknowledge for the University of Malaya under UMRG Top Down Programme, Grant No. RP027-14AET.

References

1. Mendel JM (2007) Type-2 fuzzy sets and systems: an overview. IEEE Comput Intell Mag 2(1):20–29. doi:10.1109/MCI.2007.380672
2. Zadeh LA (1974) The concept of a linguistic variable and its application to approximate reasoning. Springer, New York
3. Zadeh LA (1965) Fuzzy sets. Inf Control 8(3):338–353
4. Bellman RE, Zadeh LA (1970) Decision-making in a fuzzy environment. Manag Sci 17(4):B-141–B-164
5. Castillo O, Melin P, Kacprzyk J, Pedrycz W (2007) Type-2 fuzzy logic: theory and applications. In: IEEE international conference on granular computing (GRC 2007). IEEE, p 145
6. Mendel JM, John RIB (2002) Type-2 fuzzy sets made simple. IEEE Trans Fuzzy Syst 10(2):117–127. doi:10.1109/91.995115
7. Qilian L, Mendel JM (2000) Interval type-2 fuzzy logic systems: theory and design. IEEE Trans Fuzzy Syst 8(5):535–550. doi:10.1109/91.873577
8. Karnik NN, Mendel JM, Qilian L (1999) Type-2 fuzzy logic systems. IEEE Trans Fuzzy Syst 7(6):643–658. doi:10.1109/91.811231
9. Karnik NN, Mendel JM (1998) Introduction to type-2 fuzzy logic systems. In: IEEE world congress on computational intelligence. The IEEE international conference on fuzzy systems proceedings, 4–9 May, vol 912, pp 915–920. doi:10.1109/FUZZY.1998.686240
10. Mendel JM (2014) General type-2 fuzzy logic systems made simple: a tutorial. IEEE Trans Fuzzy Syst 22(5):1162–1182. doi:10.1109/Tfuzz.2013.2286414
11. Castillo O, Melin P (2012) Optimization of type-2 fuzzy systems based on bio-inspired methods: a concise review. Inf Sci 205:1–19. doi:10.1016/j.ins.2012.04.003
12. Fayek HM, Elamvazuthi I, Perumal N, Venkatesh B (2014) A controller based on optimal type-2 fuzzy logic: systematic design, optimization and real-time implementation. ISA Trans 53(5):1583–1591. doi:10.1016/j.isatra.2014.06.001
13. Sun Z, Wang N, Bi YR (2015) Type-1/type-2 fuzzy logic systems optimization with RNA genetic algorithm for double inverted pendulum. Appl Math Model 39(1):70–85. doi:10.1016/j.apm.2014.04.035
14. Baklouti N, Alimi AM (2013) Real time PSO based adaptive learning type-2 fuzzy logic controller design for the iRobot create robot. In: International conference on individual and collective behaviors in robotics (ICBR), 15–17 December, pp 15–20. doi:10.1109/ICBR.2013.6729284
15. Martínez-Soto R, Castillo O, Aguilar LT (2014) Type-1 and type-2 fuzzy logic controller design using a hybrid PSO-GA optimization method. Inf Sci 285:35–49. doi:10.1016/j.ins.2014.07.012
16. Chiroma H, Abdulkareem S, Herawan T (2015) Evolutionary neural network model for West Texas intermediate crude oil price prediction. Appl Energy 142:266–273
17. Chiroma H, Abdul-Kareem S, Khan A, Nawi NM, Gital AYU, Shuib L, Abubakar IA, Rahman MZ, Herawan T (2015) Global warming: predicting OPEC carbon dioxide emissions from petroleum consumption using neural network and hybrid cuckoo search algorithm. PLoS One 10(8):e0136140
18. Castillo O, Melin P (2014) A review on interval type-2 fuzzy logic applications in intelligent control. Inf Sci 279:615–631. doi:10.1016/j.ins.2014.04.015
19. Lu T-C (2015) Genetic-algorithm-based type reduction algorithm for interval type-2 fuzzy logic controllers. Eng Appl Artif Intell 42:36–44
20. Pulido M, Melin P, Castillo O (2014) Particle swarm optimization of ensemble neural networks with fuzzy aggregation for time series prediction of the Mexican stock exchange. Inf Sci 280:188–204. doi:10.1016/j.ins.2014.05.006
21. Qi-Ye Z, Ze-Ming S, Feng Z (2014) A clustering routing protocol for wireless sensor networks based on type-2 fuzzy logic and ACO. In: IEEE international conference on fuzzy systems (FUZZ-IEEE), 6–11 July, pp 1060–1067. doi:10.1109/FUZZ-IEEE.2014.6891584
22. Kumbasar T, Hagras H (2014) Big bang–big crunch optimization based interval type-2 fuzzy PID cascade controller design strategy. Inf Sci 282:277–295. doi:10.1016/j.ins.2014.06.005
23. Kiani M, Mohammadi SMA, Gharaveisi AA (2013) A bacterial foraging optimization approach for tuning type-2 fuzzy logic controller. Turk J Electr Eng Comput Sci 21(1):263–273. doi:10.3906/Elk-1107-10

24. Sayed MM, Saad MS, Emara HM, Abou El-Zahab EE (2013) Improving the performance of the Egyptian second testing nuclear research reactor using interval type-2 fuzzy logic controller tuned by modified biogeography-based optimization. Nucl Eng Des 262:294–305. doi:10.1016/j.nucengdes.2013.04.035
25. Astudillo L, Melin P, Castillo O (2013) Nature inspired chemical optimization to design a type-2 fuzzy controller for a mobile robot. In: Proceedings of the 2013 joint IFSA world congress and NAFIPS annual meeting (IFSA/NAFIPS), pp 1423–1428
26. El-Nagar AM, El-Bardini M (2014) Interval type-2 fuzzy neural network controller for a multivariable anesthesia system based on a hardware-in-the-loop simulation. Artif Intell Med 61(1):1–10. doi:10.1016/j.artmed.2014.03.002
27. Doostparast Torshizi A, Fazel Zarandi MH (2014) Alpha-plane based automatic general type-2 fuzzy clustering based on simulated annealing meta-heuristic algorithm for analyzing gene expression data. Comput Biol Med. doi:10.1016/j.compbiomed.2014.06.017
28. Nguyen Cong L, Meesad P (2013) Meta-heuristic algorithms applied to the optimization of type-1 and type 2 TSK fuzzy logic systems for sea water level prediction. In: IEEE sixth international workshop on computational intelligence & applications (IWCIA), 13–13 July, pp 69–74. doi:10.1109/IWCIA.2013.6624787
29. Liu P, Yang LX, Wang L, Li SK (2014) A solid transportation problem with type-2 fuzzy variables. Appl Soft Comput 24:543–558. doi:10.1016/j.asoc.2014.08.005
30. Melin P, Castillo O (2014) A review on type-2 fuzzy logic applications in clustering, classification and pattern recognition. Appl Soft Comput 21:568–577. doi:10.1016/j.asoc.2014.04.017
31. Melin P, Castillo O (2013) A review on the applications of type-2 fuzzy logic in classification and pattern recognition. Expert Syst Appl 40(13):5413–5423. doi:10.1016/j.eswa.2013.03.020
32. Castillo O, Melin P (2012) A review on the design and optimization of interval type-2 fuzzy controllers. Appl Soft Comput 12(4):1267–1278. doi:10.1016/j.asoc.2011.12.010
33. Sanchez MA, Castillo O, Castro JR (2015) Generalized type-2 fuzzy systems for controlling a mobile robot and a performance comparison with interval type-2 and type-1 fuzzy systems. Expert Syst Appl 42(14):5904–5914
34. Naim S, Hagras H (2013) A big-bang big-crunch optimized general type-2 fuzzy logic approach for multi-criteria group decision making. J Artif Intell Soft Comput Res. doi:10.2478/jaiscr-2014-0009
35. Kumbasar T, Hagras H (2015) A self-tuning zSlices-based general type-2 fuzzy PI controller. IEEE Trans Fuzzy Syst 23(4):991–1013. doi:10.1109/TFUZZ.2014.2336267
36. Mendel JM, John RI, Feilong L (2006) Interval type-2 fuzzy logic systems made simple. IEEE Trans Fuzzy Syst 14(6):808–821. doi:10.1109/TFUZZ.2006.879986
37. Mendel JM (1995) Fuzzy logic systems for engineering: a tutorial. Proc IEEE 83(3):345–377. doi:10.1109/5.364485
38. Wang LX, Mendel JM (1992) Fuzzy basis functions, universal approximation, and orthogonal least-squares learning. IEEE Trans Neural Netw 3(5):807–814. doi:10.1109/72.159070
39. Wang LX, Mendel JM (1992) Generating fuzzy rules by learning from examples. IEEE Trans Syst Man Cybern 22(6):1414–1427. doi:10.1109/21.199466
40. Linda O, Manic M (2012) General type-2 fuzzy c-means algorithm for uncertain fuzzy clustering. IEEE Trans Fuzzy Syst 20(5):883–897
41. Zhai D, Mendel JM (2011) Computing the centroid of a general type-2 fuzzy set by means of the centroid-flow algorithm. IEEE Trans Fuzzy Syst 19(3):401–422
42. Zhai D, Mendel JM (2011) Uncertainty measures for general type-2 fuzzy sets. Inf Sci 181(3):503–518
43. Wagner C, Hagras H (2010) Toward general type-2 fuzzy logic systems based on zSlices. IEEE Trans Fuzzy Syst 18(4):637–660
44. Wagner C, Hagras H (2008) zSlices—towards bridging the gap between interval and general type-2 fuzzy logic. In: IEEE international conference on fuzzy systems (FUZZ-IEEE 2008). IEEE world congress on computational intelligence. IEEE, pp 489–497
45. Liu F (2008) An efficient centroid type-reduction strategy for general type-2 fuzzy logic system. Inf Sci 178(9):2224–2236
46. Lucas LA, Centeno TM, Delgado MR (2007) General type-2 fuzzy inference systems: analysis, design and computational aspects. In: IEEE international fuzzy systems conference (FUZZ-IEEE 2007). IEEE, pp 1–6
47. Coupland S, John R (2007) Geometric type-1 and type-2 fuzzy logic systems. IEEE Trans Fuzzy Syst 15(1):3–15
48. Mendel JM (2007) Advances in type-2 fuzzy sets and systems. Inf Sci 177(1):84–110
49. Mendel J, Hagras H, Tan W-W, Melek WW, Ying H (2014) Introduction to type-2 fuzzy logic control: theory and applications. Wiley, New York
50. Mendel JM (2001) Uncertain rule-based fuzzy logic system: introduction and new directions. Prentice Hall PTR, Upper Saddle River
51. Castillo O (2012) Introduction to type-2 fuzzy logic control. In: Kacprzyk J (ed) Type-2 fuzzy logic in intelligent control applications. Springer, New York, pp 3–5
52. Dongrui W (2013) Approaches for reducing the computational cost of interval type-2 fuzzy logic systems: overview and comparisons. IEEE Trans Fuzzy Syst 21(1):80–99. doi:10.1109/TFUZZ.2012.2201728
53. Karnik NN, Mendel JM (1998) Type-2 fuzzy logic systems: type-reduction. In: IEEE international conference on systems, man, and cybernetics, 11–14 October, vol 2042, pp 2046–2051. doi:10.1109/ICSMC.1998.728199
54. Karnik NN, Mendel JM (1998) Introduction to type-2 fuzzy logic systems. In: IEEE world congress on computational intelligence. IEEE international conference on fuzzy systems proceedings. IEEE, pp 915–920
55. Mendel JM (2015) On type-reduction versus direct defuzzification for type-2 fuzzy logic systems. In: Tamir DE, Rishe ND, Kandel A (eds) Fifty years of fuzzy logic and its applications. Springer, New York, pp 387–399
56. Martínez-Soto R, Castillo O, Aguilar LT, Rodriguez A (2013) A hybrid optimization method with PSO and GA to automatically design type-1 and type-2 fuzzy logic controllers. Int J Mach Learn Cybern. doi:10.1007/s13042-013-0170-8
57. Krohling RA, Rey JP (2001) Design of optimal disturbance rejection PID controllers using genetic algorithms. IEEE Trans Evol Comput 5(1):78–82
58. Zhang Z, Zhang S (2012) Type-2 fuzzy soft sets and their applications in decision making. J Appl Math 2012: 608681. doi:10.1155/2012/608681
59. Hassanzadeh I, Mobayen S (2011) Controller design for rotary inverted pendulum system using evolutionary algorithms. Math Probl Eng 2011: 572424. doi:10.1155/2011/572424
60. Castillo O (2012) Type-2 fuzzy logic in intelligent control applications, vol 272. Springer, New York
61. Kumbasar T, Hagras H (2015) Interval type-2 fuzzy PID controllers. In: Kacprzyk J, Pedrycz W (eds) Springer handbook of computational intelligence. Springer, Berlin, pp 285–294. doi:10.1007/978-3-662-43505-2_18
62. Aliasghary M, Eksin I, Guzelkaya M, Kumbasar T (2013) A design methodology and analysis for interval type-2 fuzzy PI/PD controllers. Int J Innov Comput Inf Control 9(10):4215–4230
63. Kumbasar T (2014) A simple design method for interval type-2 fuzzy PID controllers. Soft Comput 18(7):1293–1304

64. Kumbasar T (2015) Robust stability analysis and systematic design of single input interval type-2 fuzzy logic controllers. IEEE Trans Fuzzy Syst. doi:10.1109/TFUZZ.2015.2471805
65. Kumbasar T (2014) Robust stability analysis of PD type single input interval type-2 fuzzy control systems. In: IEEE international conference on fuzzy systems (FUZZ-IEEE). IEEE, pp 634–639
66. Kumbasar T, Hagras H (2015) A gradient descent based online tuning mechanism for PI type single input interval type-2 fuzzy logic controllers. In: IEEE international conference on fuzzy systems (FUZZ-IEEE). IEEE
67. Mendel JM, Rajati MR (2015) Critique of "Footprint of uncertainty for type-2 fuzzy sets"[9]. Inf Sci 308:1–2
68. Sánchez D, Melin P, Castillo O (2015) Fuzzy system optimization using a hierarchical genetic algorithm applied to pattern recognition. In: Felev D, Jablkowski J, Kacprzyk J, Karawczak M, Popchev I, Rutkowski L, Sgurev V, Sotirova E, Szynkarczyk P, Zadrozny S (eds) Intelligent systems' 2014. Springer, New York, pp 713–720
69. Oh S-K, Jang H-J, Pedrycz W (2011) A comparative experimental study of type-1/type-2 fuzzy cascade controller based on genetic algorithms and particle swarm optimization. Expert Syst Appl 38(9):11217–11229. doi:10.1016/j.eswa.2011.02.169
70. Chiroma H, Abdulkareem S, Abubakar A, Mohammed JU (2013) Computational intelligence techniques with application to crude oil price projection: a literature survey from 2001–2012. Neural Netw World 23(6):523–551
71. Chaparro B, Thuillier S, Menezes L, Manach P-Y, Fernandes J (2008) Material parameters identification: gradient-based, genetic and hybrid optimization algorithms. Comput Mater Sci 44(2):339–346
72. Xia W, Wu Z (2005) An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems. Comput Ind Eng 48(2):409–425
73. Kao Y-T, Zahara E (2008) A hybrid genetic algorithm and particle swarm optimization for multimodal functions. Appl Soft Comput 8(2):849–857
74. Kim DH, Abraham A, Cho JH (2007) A hybrid genetic algorithm and bacterial foraging approach for global optimization. Inf Sci 177(18):3918–3937
75. Kim DH (2007) GA-PSO based vector control of indirect three phase induction motor. Appl Soft Comput 7(2):601–611
76. Dulikravich GS, Colaço MJ (2015) Hybrid optimization algorithms and hybrid response surfaces. In: Greiner D, Galván B, Periaux J, Gauger N, Giannakoglou K, Winter G (eds) Advances in evolutionary and deterministic methods for design, optimization and control in engineering and sciences. Springer, New York, pp 19–47
77. Haupt RL, Haupt SE (2004) Practical genetic algorithms. Wiley, New York
78. Sivanandam SN, Deepa SN (2008) Introduction to genetic algorithms. Springer, New York
79. Herrera F (2008) Genetic fuzzy systems: taxonomy, current research trends and prospects. Evol Intel 1(1):27–46
80. Cordón O (2011) A historical review of evolutionary learning methods for Mamdani-type fuzzy rule-based systems: designing interpretable genetic fuzzy systems. Int J Approx Reason 52(6):894–913
81. Cordón O, Herrera F, Gomide F, Hoffmann F, Magdalena L (2001) Ten years of genetic fuzzy systems: current framework and new trends. In: Joint 9th IFSA world congress and 20th NAFIPS international conference. IEEE, pp 1241–1246
82. De Jong K (1988) Learning with genetic algorithms: an overview. Mach Learn 3(2–3):121–138
83. Goldberg DE (2002) The design of competent genetic algorithms: steps toward a computational theory of innovation. Kluwer, Dordrecht
84. Man K-F, Tang KS, Kwong S (2012) Genetic algorithms: concepts and designs. Springer, New York
85. Martínez R, Castillo O, Aguilar LT (2009) Optimization of interval type-2 fuzzy logic controllers for a perturbed autonomous wheeled mobile robot using genetic algorithms. Inf Sci 179(13):2158–2174
86. Chiroma H, Abdulkareem S, Sari EN, Abdullah Z, Muaz SA, Kaynar O, Shah H, Herawan T (2014) Soft computing approach in modelling energy consumption. In: Murgante B, Misra S, Rocha AMA, Torre C, Rocha JG, Falcão MI, Taniar D, Apduhan BO, Gervasi O (eds) Computational science and its applications (ICCSA 2014). Springer, New York, pp 770–782
87. Eberhart RC, Kennedy J (1995) A new optimizer using particle swarm theory. In: Proceedings of the sixth international symposium on micro machine and human science, New York, NY, USA, pp 39–43
88. Dehuri S, Ghosh S, Coello CAC (2009) An introduction to swarm intelligence for multi-objective problems in data mining. In: Dehuri S, Ghosh S, Coello CAC (eds) Swarm intelligence for multi-objective problems in data mining. Springer, New York
89. Eberhart RC, Shi Y (2001) Particle swarm optimization: developments, applications and resources. In: Proceedings of the 2001 congress on evolutionary computation. IEEE, pp 81–86
90. Clerc M, Kennedy J (2002) The particle swarm—explosion, stability, and convergence in a multidimensional complex space. IEEE Trans Evol Comput 6(1):58–73. doi:10.1109/4235.985692
91. Chiou J-S, Liu M-T (2009) Numerical simulation for fuzzy-PID controllers and helping EP reproduction with PSO hybrid algorithm. Simul Model Pract Theory 17(10):1555–1565
92. Shahsadeghi M, Khooban MH, Niknam T (2014) A robust and simple optimal type II fuzzy sliding mode control strategy for a class of nonlinear chaotic systems. J Intell Fuzzy Syst 27(4):1849–1859. doi:10.3233/Ifs-141151
93. Maldonado Y, Castillo O, Melin P (2013) Particle swarm optimization of interval type-2 fuzzy systems for FPGA applications. Appl Soft Comput 13(1):496–508. doi:10.1016/j.asoc.2012.08.032
94. Martinez-Soto R, Castillo O, Aguilar LT, Rodriguez A (2015) A hybrid optimization method with PSO and GA to automatically design type-1 and type-2 fuzzy logic controllers. Int J Mach Learn Cybern 6(2):175–196. doi:10.1007/s13042-013-0170-8
95. Li CT, Lee CH, Chang FY, Lin CM (2014) An interval type-2 fuzzy system with a species-based hybrid algorithm for nonlinear system control design. Math Probl Eng. doi:10.1155/2014/735310
96. Hsu CH, Juang CF (2013) Evolutionary robot wall-following control using type-2 fuzzy controller with species-DE-activated continuous ACO. IEEE Trans Fuzzy Syst 21(1):100–112. doi:10.1109/Tfuzz.2012.2202665
97. Cervantes L, Castillo O (2015) Type-2 fuzzy logic aggregation of multiple fuzzy controllers for airplane flight control. Inf Sci 324:247–256
98. Maldonado Y, Castillo O, Melin P (2014) A multi-objective optimization of type-2 fuzzy control speed in FPGAs. Appl Soft Comput 24:1164–1174. doi:10.1016/j.asoc.2014.04.041
99. Castillo O, Cervantes L (2014) Genetic design of optimal type-1 and type-2 fuzzy systems for longitudinal control of an airplane. Intell Autom Soft Comput 20(2):213–227. doi:10.1080/10798587.2014.902913
100. Melendez A, Castillo O, Melin P (2013) Genetic optimization of interval type-2 fuzzy reactive controllers for mobile robots. In: 2013 joint IFSA world congress and NAFIPS annual meeting (IFSA/NAFIPS), 24–28 June, pp 1418–1422. doi:10.1109/IFSA-NAFIPS.2013.6608609
101. Cervantes L, Castillo O (2013) Statistical comparison of type-1 and type-2 fuzzy systems design with genetic algorithms in the

case of three tank water control. In: Proceedings of the 2013 joint IFSA world congress and NAFIPS annual meeting (IFSA/NAFIPS), pp 1056–1061
102. Bi YR, Srinivasan D, Lu XB, Sun Z (2013) Single intersection signal control based on type-2 fuzzy logic. In: Proceedings of the 2013 IEEE symposium on computational intelligence in vehicles and transportation systems (CIVTS), pp 25–31
103. Shill PC, Amin MF, Akhand MAH, Murase K (2012) optimization of interval type-2 fuzzy logic controller using quantum genetic algorithms. In: 2012 IEEE international conference on fuzzy systems (FUZZ-IEEE), 10–15 June, pp 1–8. doi:10.1109/FUZZ-IEEE.2012.6251207
104. Maldonado Y, Castillo O (2012) Comparison between multiobjective GA and PSO for parameter optimization of AT2-FLC for a real application in FPGA. In: 2012 annual meeting of the North American fuzzy information processing society (NAFIPS), 6–8 August, pp 1–6. doi:10.1109/NAFIPS.2012.6291047
105. Maldonado Y, Castillo O (2012) Genetic design of an interval type-2 fuzzy controller for velocity regulation in a DC motor. Int J Adv Robot Syst. doi:10.5772/51188
106. Li C, Zhang X, Yi J SIRMs connected type-2 fuzzy-genetic backing up control of the truck-trailer system. In: 2012 31st Chinese control conference (CCC), 25–27 July, pp 3536–3541
107. Ghaemi M, Akbarzadeh-T MR, Jalaeian-F M (2012) Adaptive interval type-2 fuzzy PI sliding mode control with optimization of membership functions using genetic algorithm. In: 2012 2nd international econference on computer and knowledge engineering (ICCKE), 18–19 October, pp 123–128. doi:10.1109/ICCKE.2012.6395364
108. Hamza MF, Yap HJ, Choudhury IA (2015) Genetic algorithm and particle swarm optimization based cascade interval type 2 fuzzy PD controller for rotary inverted pendulum system. Math Probl Eng. doi:10.1155/2015/695965
109. Niknam T, Khooban MH, Kavousifard A, Soltanpour MR (2014) An optimal type II fuzzy sliding mode control design for a class of nonlinear systems. Nonlinear Dyn 75(1–2):73–83. doi:10.1007/s11071-013-1050-1
110. Khooban MH, Abadi DNM, Alfi A, Siahi M (2014) Optimal type-2 fuzzy controller for HVAC systems. Automatika 55(1):69–78. doi:10.7305/automatika.2014.01.219
111. Allawi ZT, Abdalla TY (2014) A PSO-optimized type-2 fuzzy logic controller for navigation of multiple mobile robots. In: 2014 19th international conference on methods and models in automation and robotics (MMAR), 2–5 September, pp 33–39. doi:10.1109/MMAR.2014.6957321
112. Panda MK, Pillai GN, Kumar V (2012) Power system stabilizer design: interval type-2 fuzzy logic controller approach. In: 2012 2nd international conference on power, control and embedded systems (ICPCES 2012)
113. Castillo O, Martinez-Marroquin R, Melin P, Valdez F, Soria J (2012) Comparative study of bio-inspired algorithms applied to the optimization of type-1 and type-2 fuzzy controllers for an autonomous mobile robot. Inf Sci 192:19–38. doi:10.1016/j.ins.2010.02.022
114. Yesil E (2014) Interval type-2 fuzzy PID load frequency controller using big bang–big crunch optimization. Appl Soft Comput 15:100–112. doi:10.1016/j.asoc.2013.10.031
115. Sayed MM, Saad MS, Emara HM, Abou El-Zahab EE (2013) A novel method for type-2 fuzzy logic controller design using a modified biogeography-based optimization. In: 2013 IEEE international conference on industrial technology (ICIT), 25–28 February, pp 28–33. doi:10.1109/ICIT.2013.6505643
116. Melin P, Astudillo L, Castillo O, Valdez F, Garcia M (2013) Optimal design of type-2 and type-1 fuzzy tracking controllers for autonomous mobile robots under perturbed torques using a new chemical optimization paradigm. Expert Syst Appl 40(8):3185–3195
117. Astudillo L, Melin P, Castillo O (2012) Nature optimization applied to design a type-2 fuzzy controller for an autonomous mobile robot. In: 2012 fourth world congress on nature and biologically inspired computing (NaBIC), 5–9 November, pp 212–217. doi:10.1109/NaBIC.2012.6402264
118. Li CD, Zhang GQ, Wang M, Yi JQ (2013) Data-driven modeling and optimization of thermal comfort and energy consumption using type-2 fuzzy method. Soft Comput 17(11):2075–2088. doi:10.1007/s00500-013-1117-4
119. Cortes-Rios JC, Gomez-Ramirez E, Ortiz-de-la-Vega HA, Castillo O, Melin P (2014) Optimal design of interval type 2 fuzzy controllers based on a simple tuning algorithm. Appl Soft Comput 23:270–285. doi:10.1016/j.asoc.2014.06.015
120. Nawi NM, Khan A, Rehman M, Chiroma H, Herawan T (2015) Weight optimization in recurrent neural networks with hybrid meta heuristic cuckoo search techniques for data classification. Math Probl Eng 501:868375
121. Amador-Angulo L, Castillo O (2014) Optimization of the type-1 and type-2 fuzzy controller design for the water tank using the bee colony optimization. In: 2014 IEEE conference on Norbert Wiener in the 21st century (21CW), 24–26 June, pp 1–8. doi:10.1109/NORBERT.2014.6893876
122. Chiroma H, Shuib NLM, Muaz SA, Abubakar AI, Ila LB, Maitama JZ (2015) A review of the applications of bio-inspired flower pollination algorithm. Procedia Comput Sci 62:435–441