PSO OPTIMIZED INTERVAL TYPE-2 FUZZY DESIGN FOR ELECTIONS RESULTS PREDICTION

Uduak Umoh[1], Samuel Udoh[1], Etebong Isong[2], Regina Asuquo[1], and Emmanuel Nyoho[1]

[1]Department of Computer Science, University of Uyo, PMB 1017, Akwa Ibom State, Nigeria
[2]Department of Computer Science, Akwa Ibom State University, Mkpatenin, Nigeria

ABSTRACT

Interval type-2 fuzzy logic systems (IT2FLSs), have recently shown great potential in various applications with dynamic uncertainties. It is believed that additional degree of uncertainty provided by IT2FL allows for better representation of the uncertainty and vagueness present in prediction models. However, determining the parameters of the membership functions of IT2FL is important for providing optimum performance of the system. Particle Swarm Optimization (PSO) has attracted the interest of researchers due to their simplicity, effectiveness and efficiency in solving real-world optimization problems. In this paper, a novel optimal IT2FLS is designed, applied for predicting winning chances in elections. PSO is used as an optimized algorithm to tune the parameter of the primary membership function of the IT2FL to improve the performance and increase the accuracy of the IT2F set. Simulation results show the superiority of the PSO-IT2FL to the similar non-optimal IT2FL system with an increase in the prediction.

KEYWORDS

Type-2 Fuzzy logic system, Particle swarm optimization, optimum fuzzy membership function, politics, democracy, election prediction accuracy

1. INTRODUCTION

In a technical sense, an election is a process by which an office is assigned to a person by an act of voting that involves the simultaneous expression of opinion by many people. Elections have been the usual mechanism by which modern representative democracy has operated since the 17th century. It serves as a process of changing the government in a country through peaceful means. The process and properties may vary for different sectors and different countries. The election prediction, on the other hand, aims at forecasting the outcome of elections. The major challenge in election prediction is the task of predicting an accurate result. The goal of predicting election outcomes requires forecasters to pay attention to the political and economic indicators that vary across a series of electoral contests, such as macroeconomic conditions and popularity ratings. There are three leading approaches to Election prediction – opinion polls, prediction markets, and models [1].

Opinion polls are usually designed to represent the opinions of a population by conducting a series of questions and then extrapolating generalities in ratio or within confidence intervals. Prediction markets are betting markets where people buy and sell candidate futures based on who they think will win the election. Models use non-polling aggregate data to predict election outcomes with the use of measures of government and economic performance. All three are united in that they are driven by some methodological theory, but models represent the most accurate and reliable method of predicting elections to date [2].

DOI: 10.5121/ijfls.2019.9101

To predict an accurate result of elections with respect to the different candidates and parties is quite a challengeable task since the system is linguistic, vague, and dynamic in nature. Also predicting more accurately requires more information and the use of more parameters which has been lacking. A variety of predictive models such as statistical, regression analysis, predicting from past experiences, etc. are proposed for elections predictions and have given inaccurate results. Due to the changing times and trends, statistical methods are unable to predict elections in a realistic way sequel to their inadequacy to represent fuzzy information that characterize election processes. Traditional rigorous mathematical approaches are inappropriate for the modeling of this kind of humanistic system due to the imprecision in their linguistic expressions. In general, economic conditions and public opinion surveys are used to build the models for predicting the election outcome. [3] [4] are some typical contributions on election forecasting from political scientists. [5], an economist also developed models based on economic variables in predicting election outcomes. In numerous cases, the existing legal framework establishes that political parties should “democratically” elect their candidates, but this concept is very vague, and there are few if any applicable legal provisions.

However, the election systems are inherently vague and cannot be expressed easily in precise numbers. The vagueness is both in the values of the variables and in which variable is important and thus should be considered. For example, public opinion surveys and economic conditions are frequently expressed linguistically as “very important, important, not important” or “very good, good, not good”. These linguistic expressions cannot be converted into precise numbers without loss of some of the original meanings. The concept of information is inherently associated with the concept of uncertainty which affects decision-making due to some deficiency. Uncertain information may be incomplete, imprecise, fragmentary, not fully reliable, vague, contradictory, or deficient in some other way [6] [7] [8]. Recently, optimization techniques, inspired in social behaviours, have evolved as an alternative to statistical and regression analysis model for prediction, through use of information about real or synthetic data [9]. Some of these methods are genetic algorithms [10] [11], type-1 fuzzy logic [12] [13], type-2 fuzzy logic [7], particle swarm optimization [14] [15], etc. Basically, the interest in evolutionary approach is due to easy code building and implementation, no usage of information about gradients and, capacity to escape from local optimal [9] [16].

T1FL, developed by [12] is a form of many-valued logic based on fuzzy set theory. The initial idea of T1FL is to generalize the notion of membership to a continuous range of [0, 1], called the *membership grade,* to indicate its gradient nature, where the membership grade for each point x in the universe is another point $\mu(x) \in [0,1]$. T1FL deals with reasoning that is approximate rather than fixed and exact and helps in modeling knowledge through the use of if-then fuzzy rules. A T1FL system is made up of fuzzifier, rule-base, fuzzy inference engine, and defuzzification units. It has been applied in many scientific and engineering fields, including social science in the area of politics [17] [18] [19] [20] [21] [22] [23] [24] [25]. However, T1FL has limited capabilities to directly and adequately handle uncertainties and imprecision because the T1F set has a membership grade that is crisp.

The T2FL, which is an extension of T1FL [7], has evolved to overcome the limitation of T1FL. In T2F sets, the membership grade $\mu(x)$ is a distribution of the expected point in [0, 1]. However, it is difficult and computationally intensive estimating this distribution and to work with T2FL. The IT2FL, a special case of T2FL, attempts to capture the uncertainty of T2F sets using T1F sets. The IT2FL has all the benefits of the T1FL system. In addition, it has a type-reduction unit that generates a T1F set output. The IT2FLS is again characterized by IF-THEN rules, but its antecedent or consequent sets are now type-2.

The main idea of IT2FL in politics resides in the fact that the political expert reasoning and election data are strongly based on vague and uncertain statements so common in the social field. IT2F sets are employed in this paper because they give the theoretical benefits of full T2F sets, capable of handling uncertainties and imprecision of the parameters better than T1FL and are practical to use. Thus, it can be used when the circumstances are too uncertain to determine exact membership grades in order to improve result accuracy, tractability, robustness and low-cost solutions, as in the case of election results prediction. However, in order to find the optimal intelligent prediction of election results, the bio-inspired method is applied on the optimization of the membership functions' parameters of IT2FL to obtain the optimal one.

Particle swarm optimization (PSO), introduced by Kennedy and Eberhart [26] is a metaheuristic approach very useful in optimization problems. PSO maintains a swarm of particles and each particle represents a possible solution. These particles "fly" through a multidimensional search space, where the position of each particle is adjusted according to ones experience and that of its neighbours [27]. In this paper, an IT2FL is used to predict election results. This predictor is optimized by Particle Swarm Optimization (PSO) algorithm for an optimal result. The remainder of this paper is organized as follows; Section 2 presents related literature while section 3 gives proposed design. Results and discussions are presented in section 4. Section 5 presents the conclusion and section 6 gives references.

2. THE INTERVAL TYPE-TWO CONTROLLERS

The T1FL, developed by Professor L. A. Zadeh [12], is a form of many-valued logic and it deals with reasoning that is approximate rather than fixed and exact. FL has rapidly become one of the most successful of today's technologies for developing sophisticated control systems because of its ability to handle vague statement, uncertain or imprecise information and also in decision-making system (DMS). In recent years the T1FLS has been on increase application in many to solve real-world problems because of the fact that fuzzy systems can deal with linguistic data along with the numerical data. T1FL has been applied in scientific, industrial, healthcare, etc for the design, development and implementation of algorithms and models. T1FL design can accommodate the ambiguities of real-world in human language and logic, unlike traditional approaches which require accurate equations to model real-world behaviours.

Although T1FLS can handle the uncertainties related to imprecise data, in some cases a higher degree of precision is required. This triggered the introduction of IT2FLS concept [7], which is an extension of T1FL. An IT2F set is characterized by a fuzzy membership value for each element of this set which is a fuzzy number with an interval in [0, 1]. Such sets can deal with the uncertainties about the fuzzy membership value itself. The uncertainty in data can be represented by the footprint of uncertainty (FOU). The IT2FL theory has been applied to different fields of human existence. The IT2FL has the ability to handle uncertainty adequately than its T1FL counterpart with ability, reliability, capability and robustness.

The structure of an IT2FL as shown in Figure 1 is made up of five components; The fuzzification, rule-base, inference engine, type reduction and defuzzification units. Fuzzification maps inputs (real values) to fuzzy values. Inference engine applies a fuzzy reasoning mechanism to obtain a fuzzy output. The knowledge base contains a set of fuzzy rules, which is of the form $R^i\colon if x_1$ is$F_1^i$and ... $x_n is F_n^i$then$Y is G^i, i = 1,2, \ldots m$and a membership functions set known as the database. Type Reducer transforms a fuzzy set into a Type- 1 Fuzzy Set. The defuzzification maps one output to precise values. An interval type-2 fuzzy set (IT2 FS)$\tilde{A}$ is characterized by a membership interval in the universe of discourse X as;

$$\tilde{A} = \{((x, u), \mu_{\tilde{A}}(x, u)) | \forall x \in X, \forall u \in J_x \subseteq [0, 1]\} \quad (1)$$

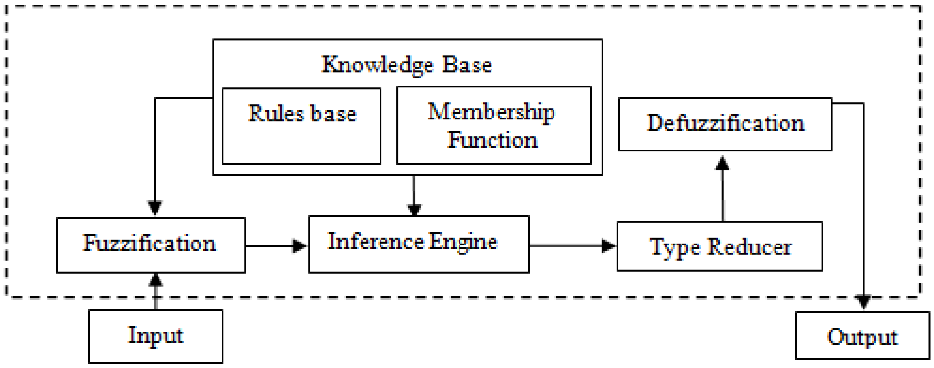

Fig. 1: Structure of an IT2FL Model for Election Results Prediction

$$\tilde{A} = \sum_{i=1}^{P}\left[\sum_{u\in Jx}[1/u]\right]/x_i \tag{2}$$

Where *x*, the *primary variable*, has domain *X*; *u*∈*U*, the *secondary variable*, has domain J_x at each *x*∈*X*;J_x is called the primary membership of x and the secondary grades of $\tilde{A}$ all equal 1. [28] [29]. Uncertainty about $\tilde{A}$ is conveyed by the union of all the primary memberships, which is called the *footprint of uncertainty* (FOU) of *A*, encompassing all the embedded primary membership functions J_x of $\tilde{A}$ as shown in 3.

$$\mu_{\tilde{A}}(x,u) = 1,\ FOU(\tilde{A}) = \cup_{\forall x\in X} J_x = \{(x,u): u \in Jx \subseteq [0,1]\} \tag{3}$$

FOU($\tilde{A}$) is bounded by *upper membership function* (UMF) $\overline{\mu}_{\tilde{A}}(x)$ and *lower membership function* (LMF) $\underline{\mu}_{\tilde{A}}(x)$, ∀*x*∈*X*, respectively assuming minimum and maximum of the membership functions of the embedded T1FSs in the FOU

$$\overline{\mu}_{\tilde{A}}(x) \equiv \overline{FOU\ (\tilde{A})} \forall x \in X \tag{4}$$

$$\underline{\mu}_{\tilde{A}}(x) \equiv \underline{FOU\ (\tilde{A})} \forall x \in X \tag{5}$$

$$\text{For IT2FS, } J_x = [\underline{\mu}_{\tilde{A}}(x), \overline{\mu}_{\tilde{A}}(x)], \forall x \in X \tag{6}$$

We employ IT2 Gaussian membership function (GMF) with fixed mean, *c* and uncertain standard deviation, σ to evaluate the degree of membership (DoM) of the input variables. GMF is suitable for a highly dynamic random system such as election results prediction.

$$\mu_A(x) = \exp\left(-\frac{x-w}{2\sigma}\right) \sigma \epsilon\ [\sigma_1, \sigma_2] \tag{7}$$

The upper and lower membership functions are calculated using:

$$\bar{\mu}_{\tilde{A}_{im}}(x_i) = \exp\left(-\frac{x_i - w_{im}}{2\bar{\sigma}^2_{2,im}}\right),\ \bar{\mu}_{\tilde{A}}(x) = N(w, \sigma_2; x) \tag{8}$$

$$\underline{\mu}_{\tilde{A}_{im}}(x_i) = \exp\left(-\frac{x_i - w_{im}}{2\underline{\sigma}^2_{1,im}}\right),\ \underline{\mu}_{\tilde{A}}(x) = N(w, \sigma_1; x) \tag{9}$$

Where *w* is the centre (mean), σ is the standard deviation and *x* is the input vector. The variables $\overline{\sigma}_{2,im}$ and $\underline{\sigma}_{1,im}$ are premise parameters that define the DoM of each element to the fuzzy set $\tilde{A}$ and FOUs of the IT2FS. The detail description is found in [30] [28]. The fuzzy rules are defined as;

$$IF x_1\ is \mu(x_1),\ \ x_2\ is \mu(x_2),\qquad AND, \ldots, AND x_m is \mu(x_m) THEN \mu_y = [\min(\mu(x_1),\ \ x_2\ is \mu(x_2), \ldots \mu(x_m))], [\max(\mu(x_1),\ \ x_2\ is \mu(x_2), \ldots \mu(x_m))] \tag{10}$$

The firing intervals for lower and upper membership functions are evaluated using [31] as;

$$F^i(x^i) = \left[\underline{\mu}_{\tilde{f}_1^i}(x_1') * \ldots * \underline{\mu}_{\tilde{f}_m^i}(x_1')\right], [\overline{\mu}_{\tilde{f}_1^i}(x_1') * \ldots * \overline{\mu}_{\tilde{f}_m^i}(x_1')] \equiv \left[\underline{f}^i, \overline{f}^i\right],\ i = 1, 2, .., M] \tag{11}$$

Where $F^i(x^i)$ is the antecedent of rule i and $\mu_{F1}{}^i(x')$ is the DoM of x in F. $\overline{\mu}_{\tilde{f}_1^i}(x)$ and $\underline{\mu}_{\tilde{f}_m^i}(x)$ are upper and lower MFs of μ_{f^i}. Instead of the product, the minimum can be used as an operator in (11). Typically, the firing intervals for low, medium and high membership functions for each input variable is evaluated as,

$$J_{\mathrm{MF}(x)} = \left[\frac{\mu_{\mathrm{FuzzyTerm}L1} + \mu_{\mathrm{FuzzyTerm}L2}}{2}, \frac{\mu_{\mathrm{FuzzyTerm}R1} + \mu_{\mathrm{FuzzyTerm}R2}}{2}\right] = \underline{\mu}_{\mathrm{FuzzyTerm}}(x_m), \overline{\mu}_{\mathrm{FuzzyTerm}}(x_m) \tag{12}$$

Where, $\mu_{\mathrm{FuzzyTerm}L1}$ and $\mu_{\mathrm{FuzzyTerm}L2}$ are the left-hand side uncertainty region boundaries and $\mu_{\mathrm{FuzzyTerm}R1}$ *and* $\mu_{\mathrm{FuzzyTerm}R2}$ are right-hand side uncertainty region boundaries for membership function variables.

The inference engine combines the fired rules and gives a mapping from input IT2FSs to output IT2FSs. Perform type-reduction for combining $F^i(x')$ and its consequent. In this paper, we explore the center of sets type-reducer (COS) method. The details are as follows:

$$Y_{TR}(x') = [y_l(x'), y_r(x')] \equiv [y_l, y_r] = \bigcup_{\substack{f^i \in F^i(x') \\ y^i \in Y^i}} \frac{\sum_{i=1}^N f^i y^i}{\sum_{i=1}^N f^i} \tag{14}$$

Which,

$$y_r = \frac{\sum_{i=1}^N f_r^i y_r^i}{\sum_{i=1}^N f_r^i} \tag{15}$$

$$y_l = \frac{\sum_{i=1}^N f_l^i y_l^i}{\sum_{i=1}^N f_l^i} \tag{16}$$

Compute output

$$y = \frac{y_l + y_r}{2} \tag{17}$$

3. THE PARTICLE SWAM OPTIMIZATION

The Particle swarm optimization (PSO) [26], is a population-based stochastic optimization technique inspired by social behaviour of bird flocking, fish schooling or human grouping to find the optimal solution using a population of particles. The fundamentals of PSO are the population of particles, interconnection topologies, search algorithms and evaluation rules which cooperate together in finding the optimal solution to the problem. The PSO searches through an n-dimensional problem space with the aim of minimizing or maximizing the objective function of the problem. In the PSO algorithm, for example, the birds in a flock are symbolically represented as "particles", considered as simple agents "flying" through a problem space. A particle's location in the multi-dimensional problem space represents one solution for the problem. When a particle moves to a new location, a different problem solution is generated. This solution is evaluated by a fitness function that provides a quantitative value of the solution's utility.

Each individual or particle of the population has both an adaptable velocity (position change) according to which it moves in the search space and a memory, remembering the best position of the search space it has ever visited. Thus, the basic concept of PSO lies in accelerating each particle towards the best individual of a topological neighbourhood with a random weighted acceleration at each time. PSO is an algorithm with a simple structure, simple parameter setting and fast convergence speed. It is widely applied to solve various function optimization problems, mathematical modeling, system control, or the problems that can be transformed to function optimization problems [32] [33] [34] [35[. Over the past years, PSO has demonstrated an efficient and successful application in many research and application areas with a better result faster and cheap as compared to other approaches. Additionally, PSO adjusts a few parameters, a slight modification of one version could well be employed in a wide variety of applications. PSO has been used to solve a wide range of problems across many applications, including, fuzzy controllers design [25] [36].

In PSO algorithm, each particle has a position vector *(Pi)* and a velocity vector (*vi*) in the search space and inertia weight *(w)*, parameter employed to control the previous velocities on the current velocity. First, the particles are initialized randomly. Then, it finds the best solution by iteration. Each particle flies through the solution space of problem and adjusts its flying velocity to search for the global optimum according to its own and social historical experiences. At every learning cycle, each particle's position and velocity are updated by the two best positions. One of which is the best solution found by the particle itself called personal best value (P*best*) and the other is the best solution found by the whole swarm, called global best value (G*best)*.
The particle velocityis updated by (18) and the position of the particle is calculated and presented in (19),while the inertia weightis calculated in (20) respectively.

$$V_i(t+1) = \alpha . V_i(t) + c_1 . r_1 . \left(p_i - X_i(t)\right) + c_2 . r_2 . (p_g - X_i(t)) \tag{18}$$

$$X_i(t+1) = X_i(t) + V_i(t+1)) \tag{19}$$

Where:

i - particle index
t - iteration number
V_i - displacement of particle's movement
$V_i(t)$ - is the current (previous) particle velocity
$V_i(t+1)$ - is the updated particle velocity
α - constriction coefficient
X_i – particle's position within the problem domain
$X_i(t)$ - current particle position
$c_1 and c_2$ - two positive constant
$r_1 and r_2$ - are normalized unit random numbers in the range [0, 1]
p_i - individual best candidate solution for particle i
p_g - global best candidate solution.

$$w = w_{max} - g$$

Where w is the inertia weight, w_{max} = 0.8 and w_{min} = 0.4, *gen* is the evolutionary generation number.

First, parameter values of i, $V_i(1)$, $X_i(1)$, $c_1$andc_2, α, p_i and p_g are initialized to begin the particle's movements within the problem space. The overall steps of the PSO algorithm are presented in Figure 2. The procedure described in Figure 2 is applied to find an optimal IT2FLCcombined with PSO to achieve a better space of solutions.

```
For each particle
 Initialize particle
End
Do
 For each particle
                Calculate fitness value
 If fitness value is better than the best fitness value (pBest) in history
                        Set current value as the new pBest
        End
Choose the particle with the best fitness value of all the particles as the gBest
        For each particle
                Calculate particle velocity
                Update particle position
        End
While maximum iteration or minimum error criteria is not attained
```

Fig. 2: The PSO Algorithm

4. THE PARTICLE INTERVAL TYPE-2 FUZZY LOGIC DESIGN FOR ELECTION RESULTS PREDICTION

Standard The IT2FL model is designed to predict the election results. The concept of IT2FL inference system is used to calculate the percentage of chances of selection of a candidate to win the election. Fuzzy Inference Systems (FIS) use fuzzy sets and "if-then" rules relevant to fuzzy sets to make decisions about incomplete or vague information. There are basically Mamdani and Sugeno, two most commonly fuzzy inference systems are present in MATLAB. The study employs Mamdani type of IT2FIS to evaluate the chances of selection of a candidate. IT2FIS system executes in five major steps: fuzzification, rulebase, inference, type-reduction and defuzzification as discussed in section 2.

In this study, initially, thirteen input parameters are collected as the factors according to their level of importance for a candidate to win an election. After proper analysis with some experts and removing/grouping some dependent arguments, only six input parameters are selected and use as input variables. The study considers popularity, the strength of the political party, credibility from past performance, financial/economic power, number of years in active politics and educational achievement as input parameters while; winning chance is the output variable, representing the level of chances of a candidate winning an election. The universe of discourse (UOD) for the inputs and output variables and the domain intervals of the variables used in the developed fuzzy models as well as the range of each variable are defined and shown in Table 1. These are partitioned according to their lower and upper values used in controlling the models. Fuzzy sets of the inputs and output variables and their associated values and labels are defined and presented in Table 2 respectively.

Table 1: Domain Intervals of Input and Output Variables

Variables	Lower Bound	Upper Bound
Popularity The strength of partyCredibility Financial State Years of Active ServiceEducational Achievement	0 0 0 0 0 0	10 1 100 10 10 8
WinningChances	0	100

Table 2: Inputs and Outputs MF Variables Fuzzy sets

Fuzzy Set	Gaussian MF Range			
	Lower σ_1	Upper σ_2	Center c	Symbol
Popularity				
Low Medium High	1.566, 0.93 1.347	2.14 1.36 1.908	0.0 5.0 10.0	LO ME HI
Strength of Party				
Weak Average Strong	0.121 0.069 0.089	0.17 0.11 0.137	0.0 0.5 1.0	LO ME HI
Credibility				
Low Moderate High	11.56 6.21 10.19	16.89 10.12 15.39	0.0 50.0 100.0	LO ME HI
Years of Active Politics				
Low Moderate High	0.936 0.708 1.06	1.443 1.135 1.607	0.0 5.0 10.0	LO ME HI
Financial/Economic Power				
Weak Average Strong	1.019 0.779 0.923	1.539 1.241 1.402	0.0 5.0 10.0	LO ME HI
Educational Achievement				
Low Moderate High	0.957 0.59 0.848	1.4 0.918 1.286	0.0 4.0 8.0	LO ME HI
Winning Chances				
Very Low Low Moderate High Very High	10.2 8.49 6.702 8.26 9.36	15.06 12.03 10.02 11.73 13.6	0.0 33.2 50.0 66.7 100.0	VL LO Mo HI VH

Fuzzification: this module maps the crisp values to interval type-2 fuzzy sets (IT2FSs) using a defined Gaussian membership function method. Each input variable has three membership functions as, *low, moderate, high*. The output variable (the level of chances of winning) has five membership functions as, *very low, low, moderate, high* and *very high*. The linguistic variable of six input memberships represented in Table 2 can be presented as follows:

Input Variable (Popularity)

$[\mu_{PopularityL1}]$, $[\mu_{PopularityL2}]$I.e. Input *Popularity* Upper and Lower membership function for Low

$[\mu_{PopularityM1}]$, $[\mu_{PopularityM2}]$ i.e. Input *Popularity* Upper and Lower membership function for Moderate

$[\mu_{PopularityH1}]$, $[\mu_{PopularityH2}]$I.e. Input *Popularity* Upper and Lower membership function for High

***Input Variable** (Strength of Party)*

$[\mu_{StrengthofPartyL1}]$, $[\mu_{StrengthofPartyL2}]$i.e. Input *Strength of Party* Upper and Lower membership function for Low

$[\mu_{StrengthofPartyM1}]$, $[\mu_{StrengthofPartyM2}]$ i.e. Input $StrengthofParty$ Upper and Lower membership function for Moderate

$[\mu_{StrengthofPartyH1}]$, $[\mu_{StrengthofPartyH2}]$i.e. Input *Popularity* Upper and Lower membership function for High

Input Variable (Strength of Party)

$[\mu_{CredibilityL1}]$, $[\mu_{CredibilityL2}]$I.e. Input $Credibility$ Upper and Lower membership function for Low

$[\mu_{CredibilityM1}]$, $[\mu_{CredibilityM2}]$ i.e. Input $Credibility$ Upper and Lower membership function for Moderate

$[\mu_{CredibilityH1}]$, $[\mu_{CredibilityH2}]$I.e. Input $Credibility$ Upper and Lower membership function for High

The same applies to financial state/economic power, years in active politics and educational achievement respectively.

Output Variable (Winning Chances)

$[\mu_{WinningChancesL1}]$, $[\mu_{WinningChancesL2}]$I.e. Input $WinningChances$ Upper and Lower membership function for Very Low

$[\mu_{WinningChancesL1}]$, $[\mu_{WinningChancesL2}]$I.e. Input $WinningChances$ Upper and Lower membership function for Low

$[\mu_{WinningChancesM1}]$, $[\mu_{WinningChancesM2}]$ i.e. Input $WinningChances$ Upper and Lower membership function for Moderate

$[\mu_{WinningChancesH1}]$, $[\mu_{WinningChancesH2}]$I.e. Input $WinningChances$ Upper and Lower membership function for High

$[\mu_{WinningChancesH1}]$, $[\mu_{WinningChancesH2}]$I.e. Input $WinningChances$ Upper and Lower membership function for $[\mu_{WinningChancesH1}]$, $[\mu_{WinningChancesH2}]$ i.e. Input $WinningChances$ Upper and Lower membership function for Very High.

Fuzzy logic toolbox in Matlab 7.5.0 is employed for the input and output membership function plots. Fuzzy rules are defined for the rules knowledge base which holds the rules employed by the inference engine. In this study, the antecedents and consequents are described using interval type-2 fuzzy sets and the rule base is created by assigning each value of the six input variables, to its maximum membership class. Parts of the fuzzy rules are presented in Table 3. Each MF of the antecedent part is represented by an upper and a lower membership function. In the IT2FLS, the rule base part is enclosed with six antecedents which are, *Popularity, Strength of Party, Credibility, Financial/Economic power, Years in active politics and Educational Achievement*. This divides the input space into a set of fuzzy regions with one consequent part (Winning Chances) which describes the system behaviour in those regions. Each of the input parameters has 3 membership functions, resulting to 3^5=243 possible combination of if-then rules in (10). Applying the expert's knowledge has reduced the number of rules to 99 rules.

Inference Engine combines the rules in a rule base and input IT2-FSs from fuzzification to produce output IT2 FSs. The firing strength *i*th rule is evaluated using the Mamdani fuzzy inference engine approach as presented in (11). A type-reduction is computed by combining the output IT2F sets and then performs a center-of-set calculation using the iterative Karnik-Mendel (KM) algorithm in (14) to produce an interval T1 FS (type-reduced set) determined by its two endpoints, y_L and y_R . Finally, we perform defuzzification by finding the average of the two endpoints,y_l and y_r in (17) and obtain the crisp value (Winning Chance). After we obtain the IT2FLC design, we then apply PSO technique to find the parameters of the IT2FLC of each candidate.

5. THE PSO OPTIMIZED INTERVAL TYPE-2 FUZZY LOGIC DESIGN FOR ELECTION RESULTS

In order to design the optimized IT2FL for election result prediction presented in section 4 above, the PSO algorithms are applied to search globally optimal parameters of primary membership functions (MFs) of the IT2FL to improve the performance and increase the accuracy of the IT2F set. The main idea consists in using the PSO algorithm to dynamically adjust the MFs of the IT2FL election predictor. The structure of the optimized IT2FL controller with PSO is presented in Figure 3.

In this paper, four the linguistic input variables are used which include; *Popularity, Strength of political party, Credibility from Past performance, Financial/Economic power, Number of years in Active politics* and *Educational Achievement*. The IT2FL-PSO algorithm for election result prediction is shown in Figure 4.The velocity of the particle is updated and the position of the particle is calculated using (18) and (19) while the inertia weight is computed using (20) respectively. The IT2FL-PSO parameters are given in Table 3. From Figure 4, the PSO algorithm is used to train the IT2FLCparameters to extract the best values that aid the prediction of election results by dynamically adjusting the MFs of the IT2FL system. The best individual/particle is injected into the population of the worst method in four iterations and vice versa. PSO uses the population/swarm in the iteration in obtaining the best individual/particle. The maximum number of iterations/generations is used in stopping the algorithm while the best result obtained from the IT2FL-PSO process is kept. The IT2FL system generating the latest *gbest* is the optimal IT2FL system. Table 4 shows the inputs and outputs MF variables fuzzy sets tuned with the PSO algorithm.

Table 3: IT2FL-PSO Parameters

S/N	Parameter	Description	Value
1	w	Inertia	1.2
2	c_1	Constant	2
3	c_2	Constant	2
4	r_1	Random number	[0,1]
5	$V_i(t)$	Particle Velocity	
6	$V_i(t+1)$		

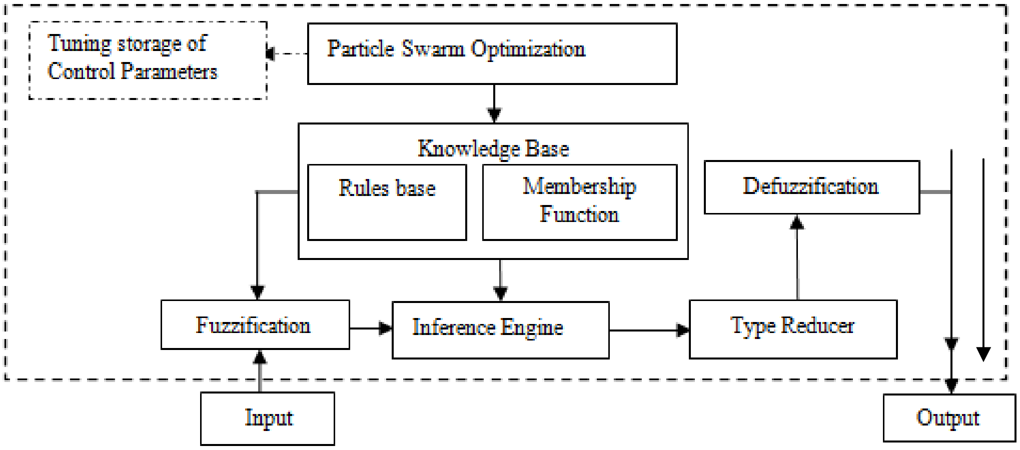

Fig. 3: PSO Optimized IT2FL for Election Result Prediction

- Each swarm of particle which is the IT2FL system is initialized. Initial velocities of all particles are randomly generated. Each IT2FL system's fitness value is calculated to find the best value, *pbest*. Set current value as the new *pbest*
- Particle velocity is updated using (18) while checking the maximum velocity.
- Set the number of iterations equal to N which is determined based on the election prediction specifications.
- Using the updated velocities, update particle position using (19).
- Update of *pbest* and *gbest*.
- Stopping criteria. While maximum iteration or minimum error criteria is not attained
- Save the best particle information

Fig. 4: IT2FL-PSO Algorithm for Election Results Prediction

Table 4: Input and output MF variable fuzzy sets after tuning with PSO algorithm.

Fuzzy Set	Gaussian MF Range			
	Lower σ_1	Upper σ_2	Center c	Symbol
Popularity				
Low	1.19	1.64	0.0	LO
Medium	1.02	1.43	5.5	ME
High	1.02	1.46	10.0	HI
Strength of Party				
Weak	0.11	0.15	0.0	LO
Average	0.11	0.16	0.5	ME
Strong	0.08	0.12	1.1	HI
Credibility				
Low	10.7	16.02	0.0	LO
Moderate	9.11	14.15	47.3	ME
High	14.4	20.24	100.0	HI

Years of Active Politics				
Low Moderate High	1.24 0.7 1.35	1.68 1.14 2.02	0.0 4.77 10.0	LO ME HI
Financial/Economic Power				
Weak Average Strong	1.01 1.08 1.35	1.53 1.6 1.99	0.0 5.24 10.0	LO ME HI
Educational Achievement				
Low Moderate High	1.24 0.68 0.84	1.68 1.01 1.28	0.0 4.36 8.0	LO ME HI
Winning Chances				
Very Low Low Moderate High Very High	10.2 8.49 6.702 8.26 9.36	15.06 12.03 10.02 11.73 13.6	0.0 33.2 50.0 66.7 100.0	VL LO Mo HI VH

6. SIMULATION RESULTS

In this paper, IT2FL for election result prediction system is developed and is applied to predict the chances of a candidate winning an election in Akwa Ibom State, Nigeria. Input data are generated based on the six variables; *Popularity, Strength of party Credibility, Financial State, Years of Active Service and Educational Achievement* while*Winning Chances* is the desired output. For each input, Gaussian membership functions with fixed mean and uncertain standard deviation are used. Secondly, we optimize IT2FL controllers using PSO approach. In this case, we combine IT2FL and PSO in order to obtain the best design of an optimal IT2FL and apply to prediction problem.

6.1 ELECTION RESULTS PREDICTION OBTAINED IT2FLC APPROACH

Figure 6 shows membership function evaluation based on the six input variables. The input and output membership functions plots showing the degree of membership are shown in Figures 6(a)-(g) respectively. Figure 7 gives theIT2FL rule viewers of chances of winning an election while Table 4 presents the results of IT2FL for Election Winning Chance. Figure 9 shows the graph of the result of IT2FL for election result prediction in Table 4.

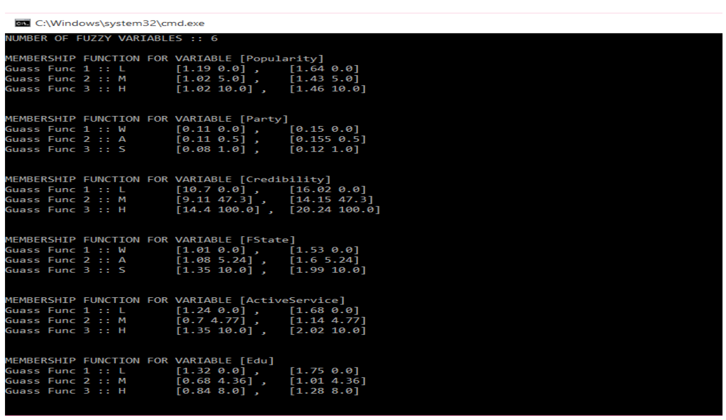

Fig. 5: Membership Function Evaluation

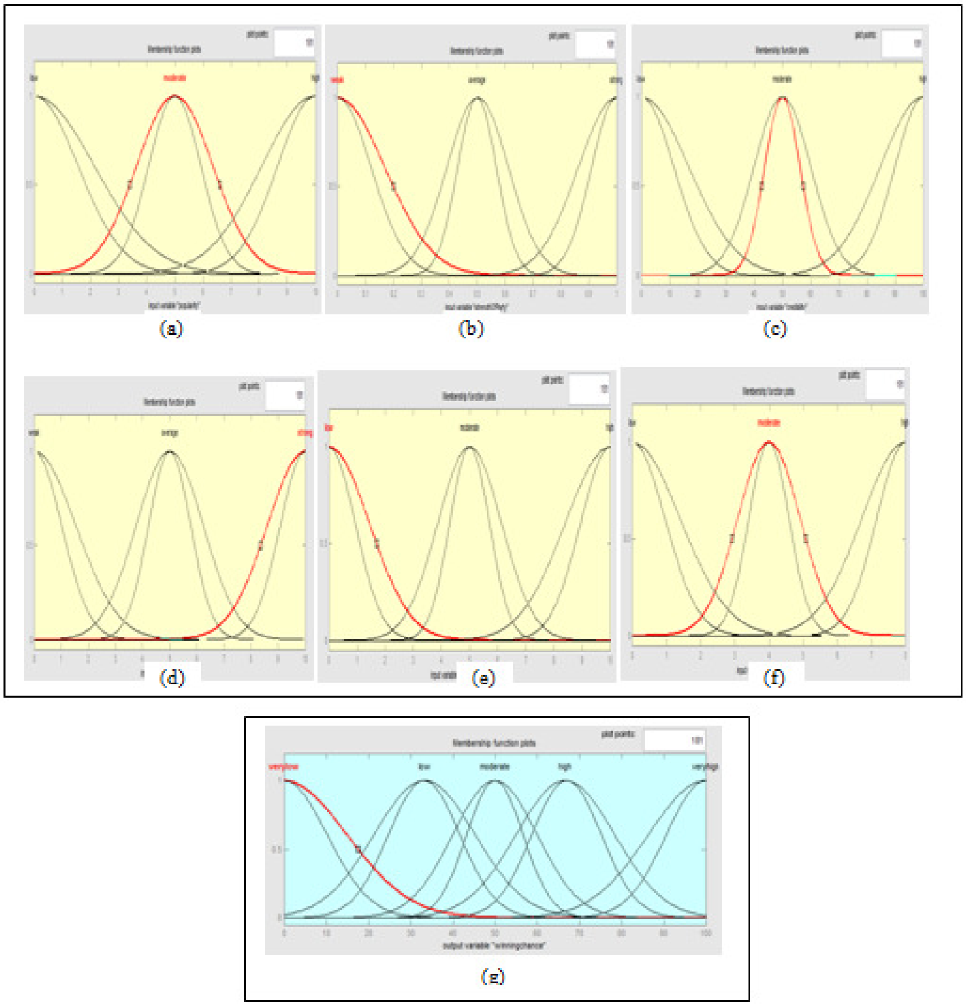

Fig. 6: IT2FL Input and output Membership function plots (a) Popularity (b) Strength of Party (c) Credibility (d) Financial /Economic power (e) Years in active politics (f) Educational achievement and (g) Winning Chances

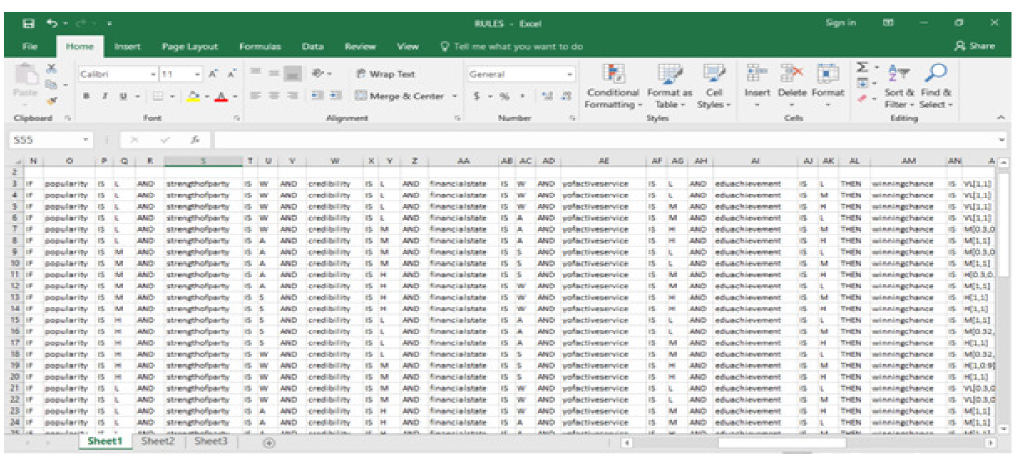

Fig. 7: IT2FL Rule Viewers of Chances of Winning Election

Table 5: Results of IT2FLC for Election Winning Chance

S/N	Popularity	Party Strength	Credibility	Financial State	Years In Active Service	Educational Background	IT2FL Result
1	3.287	0.1191	44.02	3.599	9.988	0.622	60.3
2	4.26	0.6198	57.5	9.646	1.352	1.783	60.19783
3	4.566	0.3601	25.42	4.326	7.313	5.402	50.18605
4	1.377	0.3948	10.63	1.252	1.559	2.844	17.64278
5	8.914	0.9894	17.94	5.59	1.324	2.357	60.22269
6	1.932	0.5368	88.9	6.428	2.404	5.295	78.73049
7	2.529	0.2971	44.14	2.332	8.709	5.267	59.20954
8	5.58	0.7158	45.64	3.007	8.791	7.224	75.47522
9	7.297	0.1661	76.42	3.824	1.163	6.371	60.31628
10	0.7413	0.4381	20.95	9.371	9.813	3.061	60.3
11	4.094	0.1579	46.85	9.07	3.114	4.328	58.04733
12	3.03	0.4904	15.33	3.274	9.937	0.6827	60.3
13	1.633	0.4285	84.99	3.644	9.557	7.421	80.30631
14	9.965	0.8565	31.99	7.128	6.748	6.472	80.17248
15	4.55	0.8223	79.9	8.309	9.826	2.517	60.3
20	4.286	0.5861	93.62	4.334	4.575	7.818	66.39356
21	1.669	0.4496	20.29	3.168	8.152	3.92	49.23464
.	…..	……	…..	…..	…..	……	………
39	3.581	0.9174	26.04	5.912	9.438	2.185	60.038
40	1.337	0.646	74.42	5.861	7.125	2.63	60.35297

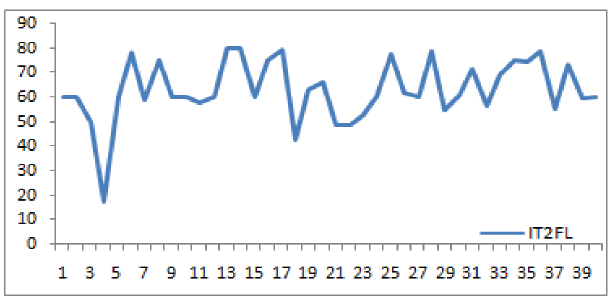

Fig. 9: The graph of the results of IT2FL for election result prediction

6.2 Optimal IT2FLC Results Obtained Using PSO Method

In this section, we present simulations results of the IT2FL controllers obtained with the PSO method. Figure 10(a-g) presents the IT2FL input and output membership function plots after tuning with the PSO algorithm.Table 5 shows theresults of IT2FL-PSO election winning chance.Figure 11 shows the graph oftheresults of IT2FL-PSO election winning chance. Table 6 contains the configuration values of the IT2FL-PSO, the execution time of hybrid IT2FL-PSO and the average error for each configuration. The goal of using the optimal controller is to obtain the best steady state error.Figure 12 shows the behaviour of theindividuals/particles of the PSO approach giving the best IT2FLC to predict the election result.The first row shows the best IT2FLC obtained by PSO with 0.0128 minimum errors.

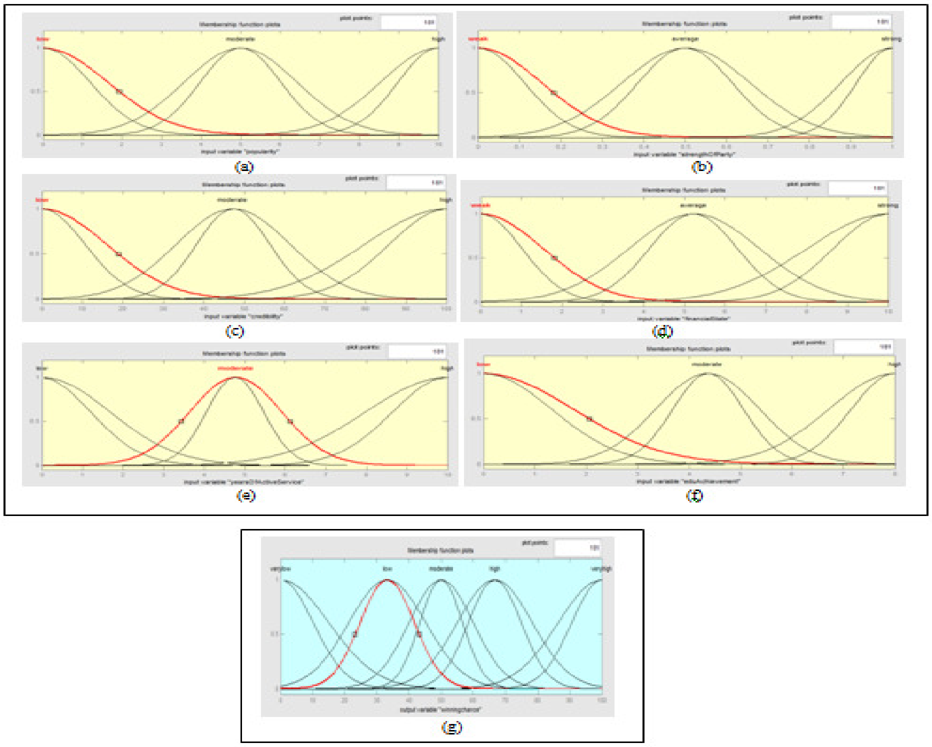

Fig. 10: IT2FL Input and output Membership Function plots after tuning with PSO (a) Popularity (b) Strength of Party (c) Credibility (d) Financial/Economic power (e) Years in active politics (f) Educational achievement and (g) Winning Chances

Table 6: IT2FL-PSO Election Winning Chance Results

S/N	Popularity	Party Strength	Credibility	Financial State	Years In Active Service	Educational Background	IT2FL-PSO
1	3.287	0.1191	44.02	3.599	9.988	0.622	62.28683
2	4.26	0.6198	57.5	9.646	1.352	1.783	62.25202
3	4.566	0.3601	25.42	4.326	7.313	5.402	52.81921
4	1.377	0.3948	10.63	1.252	1.559	2.844	21.93618
5	8.914	0.9894	17.94	5.59	1.324	2.357	62.03462
6	1.932	0.5368	88.9	6.428	2.404	5.295	79.79031
7	2.529	0.2971	44.14	2.332	8.709	5.267	64.85969
8	5.58	0.7158	45.64	3.007	8.791	7.224	76.99285
9	7.297	0.1661	76.42	3.824	1.163	6.371	63.44625
10	0.7413	0.4381	20.95	9.371	9.813	3.061	62.76658
11	4.094	0.1579	46.85	9.07	3.114	4.328	62.83292
12	3.03	0.4904	15.33	3.274	9.937	0.6827	62.28683
13	1.633	0.4285	84.99	3.644	9.557	7.421	81.90493
14	9.965	0.8565	31.99	7.128	6.748	6.472	82.07171
15	4.55	0.8223	79.9	8.309	9.826	2.517	62.28683
16	7.272	0.4901	90.98	3.748	7.557	7.41	78.17228
17	6.76	0.8196	81.73	0.6393	7.219	5.118	80.78207
18	1.244	0.6053	24.38	3.752	3.792	4.546	46.12043
19	2.837	0.1424	73.3	4.994	9.719	4.793	67.74694
.....	.....	.....	......	......	.....	.....	....
39	3.581	0.9174	26.04	5.912	9.438	2.185	62.3788
40	1.337	0.646	74.42	5.861	7.125	2.63	62.32766

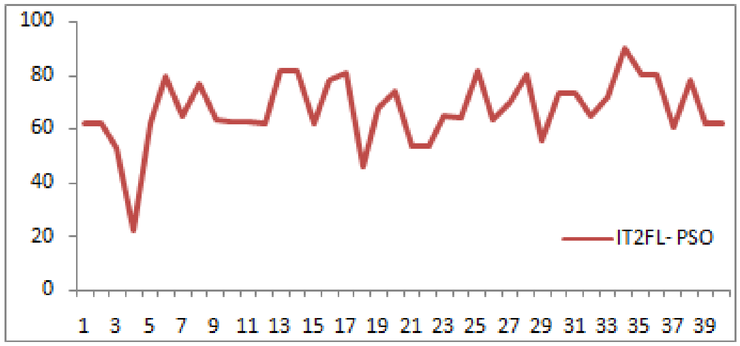

Fig. 11: The graph of the results of IT2FL-PSO for election result prediction

Table 7: Results of the IT2FL obtained by PSO approach with Execution Time and Average Error for Election Prediction

No.	Particles	Iteration/ Generation	C1	C2	Inertia	Execution Time	Error
1	200	200	0.84	0.77	0.33980	4:48:34	0.0128
2	60	70	1.2	0.94	0.49484	4:42:04	0.0411
3	50	100	2	2	0.36603	4:29:30	0.4344
4	150	45	0.58	0.97	0.63619	4:43:39	0.5412
5	70	80	1.3	1.56	0.54752	4:35:03	0.5991
6	40	75	0.63	0.79	0.47059	4:40:30	0.0834
7	25	80	1.4	0.98	0.44752	4:38:15	0.0985
8	200	70	0.81	0.91	0.13141	3:33:46	0.1232
9	90	50	1.76	1.85	0.60501	4:45:48	1.9758
10	200	80	0.73	0.15	0.84235	3:48:34	0.12

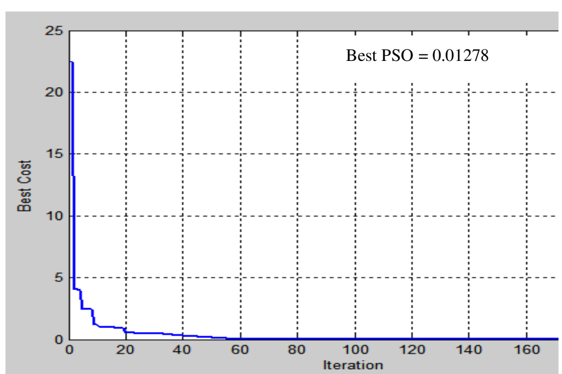

Fig. 12: The behaviour of the individuals/particles of the PSO approach with optimized IT2FL at the Best PSO of 0.01278

The comparison of IT2FL and IT2FL-PSO methods is shown in the summary of the results in Table 7. Figure 12 shows the graph of the comparison of the IT2FL andIT2FL-PSOapproaches in election result prediction.

.

Table 8: Comparison of the IT2FL andIT2FL-PSOmethods in election result prediction

No	IT2FL	IT2FL-PSO
1	60.3	62.28683
2	60.19783	62.25202
3	50.18605	52.81921
4	17.64278	21.93618
5	60.22269	62.03462
6	78.73049	79.79031
7	59.20954	64.85969
8	75.47522	76.99285
9	60.31628	63.44625
10	60.3	62.76658
11	58.04733	62.83292

No	IT2FL	IT2FL-PSO
12	60.3	62.28683
13	80.30631	81.90493
14	80.17248	82.07171
15	60.3	62.28683
16	75.36295	78.17228
17	79.85012	80.78207
18	43.09218	46.12043
19	63.61258	67.74694
20	66.39356	74.3621
…..	……	…
39	60.038	62.3788

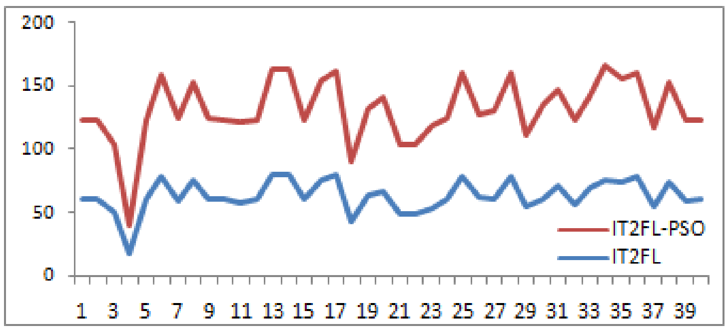

Figure 13: The graph of the comparison of the IT2FL andIT2FL-PSOapproaches in election result prediction

7. Conclusion

In this paper, we describe the application of a bio-inspired method to design an optimized IT2FL controller using PSO method. To test the optimized IT2FLC,a prediction of winning chances in an election was carried out, using a case study of Akwa Ibom State in Nigeria. To achieve our objective, IT2FL system was used which considers all the important parameters that must affect the prediction (winning chances) of a candidate in an election. Six parameters were defined using the Gaussian membership functions approach. 99 rules were defined based on "if-then" conditions and stored in the rule-base. The results of both IT2FLC and IT2FL-PSO with respect to election prediction were presented. From the IT2FL-PSO results, it was observed that the behaviour of the individuals/particles of the PSO approach with optimized IT2FL performed best at PSO value of 0.01278.

To perform a comparison of the optimization method we present a final table of results, where the average error of 10 tests of each IT2FL controller was applied to predict the winning chances in an election. The plots of the result showed that the optimal IT2FLC could get stability in less than 10 seconds. The IT2FLC obtained by PSO performed better than the result obtained with ordinary IT2FLCs in terms of stability and average error.

Generally, the simulated results indicate that the IT2FLC obtained using PSO approach as applied to election prediction system has improved the results with less error and better prediction. With the satisfactory results, the study can be used as an automatic prediction evaluator in the field of election and other related processes. In the future, the number of input parameters could be added to the system to achieve more accurate results. Hybrid approach could also be employed to improve the fuzzy logic controller design.

REFERENCES

[1] Lewis-Beck, M. S. & Tien, C. (2011). Political and election forecasting. In M. Clements and D. Hendry (Eds.), The Oxford handbook of economic forecasting (pp. 655-672).

[2] Lewis-Beck, M. S. (2005). Election forecasting: Principles and practice. British Journal of Politics and International Relations, 7, 145-164

[3] Campbell, J.E and Garand, J.C. (Eds.) (2000). Before the Vote: Forecasting American National Election, Sage Publication, Inc.

[4] Lewis-Beck, M.S and Rice, T.W. (1992). Forecasting Elections, Congressional Quarterly Press, Washington DC.

[5] Fair, R. (1988). The effects of economic events on votes for president: Update, Political Behavior 10, 168-179

[6] Castillo, O and P. Melin (2008), Type-2 Fuzzy Logic: Theory and Applications (Springer, Heidelberg, 2008)

[7] Zadeh, L.A. (1975), The concept of a linguistic variable and its application to approximate reasoning. Inf. Sci. 8, 43–80.

[8] Mendel, J.M. (2000) Uncertainty, fuzzy logic, and signal processing. Sig. Process. J. 80, 913–933.

[9] Lobato, F. S. & Souza, D. L. (2008). Adaptive Differential Evolution Method Applied To Controllers Tuning. 7th Brazilian Conference on Dynamics, Control and Applications.

[10] Bandyopadhyay, R.;Chakraborty, U. K. & Patranabis, D. (2001), Auto Tuning a PID Controller:A Fuzzy-Genetic Approach. Journal of Systems Architecture, 47(7), 663–73.

[11] Pan, I.; Das, S. & Gupta, A. (2011). Tuning of an Optimal Fuzzy PID Controller with Stochastic Algorithms for Networked Control Systems with Random Time Delay, ISA Transactions, Vol. 50, 28–36.

[12] Zadeh, L. A. (1965). Fuzzy sets: Information and Control. 8, 338-353.

[13] Hamid, B.; Mohamed, T.; Pierre-Yves, G. & Salim, L. (2010). Tuning Fuzzy PD and PI Controllers using Reinforcement Learning. ISA Transactions, Vol. 49 (4), 543–51.

[14] Kim, T-H.; Maruta, I. & Sugie, T. (2008). Robust PID Controller Tuning based on the Constrained Particle Swarm Optimization, Automatica, Vol. 44, 1104–1110.

[15] Solihin, M. I.; Tack, L. F. & Kean, M. L. (2011). Tuning of PID Controller Using Particle Swarm Optimization (PSO). Proceeding of the International Conference on Advanced Science,Engineering and Information Technology, ISBN 978-983-42366-4-9.

[16] Whiteley, P., Sanders, D., Stewart, M. & Clarke, H. (2010). Aggregate level forecasting of the 2010 general elections in Britain: The seats-votes model. Electoral Studies, 30, 278-283.

[17] Jang, J., Sun, C., Mizutani, E.: Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence. Prentice-Hall, Upper Saddle River (1997)

[18] Vaidehi, V, Monica .S, Mohamed Sheik Safeer .S, Deepika .M, Sangeetha .S A, (2008) Prediction System Based on Fuzzy Logic. Proceedings of the World Congress on Engineering and Computer Science (WCECS), October 22 - 24, 2008, San Francisco, USA

[19] Teran, L. (2011). A Fuzzy-based advisor for election and creation of political communities, Information System Research Group. 3,

[20] Singh, H, Singh, G. and Bhatia, N. (2012), Election Results Prediction System based on Fuzzy Logic. International Journal of Computer Application, 53 (9), 30-37.

[21] Mahadev, M. M. and Kulkarni, R.V. (2013, Review: Role of Fuzzy Expert System for Prediction of Election Results. Reviews Literature, 1 (2), 1-7.

[22] Pal, K. and Tyagi, S. (2014) Selection of Candidate by Political Parties Using Fuzzy Logic. International Conference on Advanced Research and Innovation (ICARI), 387-390.

[23] Umoh, U. A. and Udosen, A. (2014), Sugeno-Type Fuzzy Inference Model for Stock Price Prediction.International Journal of Computer Applications (0975 – 8887) 103 (3), 1-12.

[24] Rehman, R. (2017), Fuzzy Rule-Based Candidate Selection Evaluator by Political Parties. International Journal of Advanced Research in Computer Science, 8(3), 445-451.

[25] Umoh, U. A. and Asuquo, D. (2017) Fuzzy Logic-Based Quality of Service Evaluation for Multimedia Transmission over Wireless Ad Hoc Networks. International Journal of Computational Intelligence and Applications (IJCIA). 16(4), 1-22. DOI: 10.1142/S1469026817500237.

[26] Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, IV, pp. 1942–1948. IEEE Service Center, Piscataway (1995)

[27] Kennedy, J., Eberhart, R.: Swarm Intelligence. Morgan Kaufmann, San Francisco (2001)

[28] Mendel, J. (2001) Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions, 1st ed. Prentice-Hall, Upper Saddle River, NJ, 2(1).

[29] Karnik, N. N. and Mendel, J. M., (2001). Operations on Type-2 Fuzzy Sets, Fuzzy Sets and Systems, Vol. 122, No. 2, 2001, pp. 327-348. doi:10.1016/S0165-0114(00)00079-8

[30] Liang, Q. and N. N. Karnik (2000), Connection Admission Control in ATM Networks Using Survey-Based Type-2 Fuzzy Logic Systems'. IEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews, vol. 30, no. 3., pp. 329-340.

[31] Hagras, H. (2007). Type-2 FLCS: A new generation of fuzzy controllers. IEEE Computational Intelligence Magazine, 2(1), 30–43.

[32] Sharifi, A.; Harati, A.; Vahedian, A. Marker-based human pose tracking using adaptive annealed particle swarm optimization with search space partitioning. Image Vis. Comput. 2017, 62, 28–38

[33] Gandomi, A.H.; Yun, G.J.; Yang, X.S.; Talatahari, S. Chaos-enhanced accelerated particle swarm optimization. Commun. Nonlinear Sci. Numer. Simul. 2013, 18, 327–340.

[34] Nobile, M.S.; Cazzaniga, P.; Besozzi, D.; Colombo, R. Fuzzy self-turning PSO: A settings-free algorithm for global optimization. Swarm Evol. Comput. 2017

[35] Rao, R.V. Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. Int. J. Ind. Eng. Comput. 2016, 7, 19–34.

[36] J. Kennedy and R. Mendes, (2002), Population structure and particle swarm performance, in Proceedings of IEEE Conference on Evolutionary Computation, 2002, 1671–1676.