DE GRUYTER OPEN

JAISCR, 2018, Vol. 8, No. 2, pp. 103 – 110
10.1515/jaiscr-2018-0007

PARTICLE SWARM OPTIMIZATION FOR SOLVING A CLASS OF TYPE-1 AND TYPE-2 FUZZY NONLINEAR EQUATIONS

Sheriff Sadiqbatcha[1], Saeed Jafarzadeh[1], and Yiannis Ampatzidis[2]

[1]*Dept. of Computer and Electrical Engineering*
California State University, BakersfieldBakersfield, CA, U.S.A.

[2]*Dept. of Physics and Engineering*
California State University, BakersfieldBakersfield, CA, U.S.A.

Submitted: 30ty January 2017; Accepted: 31st March 2017

Abstract

This paper proposes a modified particle swarm optimization (PSO) algorithm that can be used to solve a variety of fuzzy nonlinear equations, i.e. fuzzy polynomials and exponential equations. Fuzzy nonlinear equations are reduced to a number of interval nonlinear equations using alpha cuts. These equations are then sequentially solved using the proposed methodology. Finally, the membership functions of the fuzzy solutions are constructed using the interval results at each alpha cut. Unlike existing methods, the proposed algorithm does not impose any restriction on the fuzzy variables in the problem. It is designed to work for equations containing both positive and negative fuzzy sets and even for the cases when the support of the fuzzy sets extends across 0, which is a particularly problematic case.

Keywords: type-1 and type-2 fuzzy sets; polynomial and exponential equations; particle swarm optimization.

1 Introduction

The mathematical space of crisp nonlinear equations is a very well-studied field. However, in many engineering applications it is common to encounter variables that cannot be described with crisp quantities. Various methods of quantifying these uncertainties have been developed over the years, e.g. intervals and fuzzy sets. However, solving equations, especially non-linear equations, containing these uncertain variables is not trivial. In this paper we propose a Particle Swarm Optimization (PSO) algorithm for solving a family of type-1 and interval type-2 fuzzy nonlinear equations. These equations include fuzzy polynomials like Eq.(1) and exponential equations like Eq. (2).

$$\tilde{a}\tilde{x}^n+\tilde{b}\tilde{x}^{n-1}+\cdots+\tilde{c}\tilde{x}+\tilde{d}=\tilde{z}, \tag{1}$$

$$\tilde{a}\exp(\tilde{x}+\tilde{b}) = \tilde{z}, \tag{2}$$

where $\tilde{a}$, $\tilde{b}$, $\tilde{c}$, $\tilde{d}$, and $\tilde{z}$ are known convex fuzzy quantities and $\tilde{x}$ is the unknown fuzzy variable. To our knowledge there are very few results on solving these types of equations, even though applications for these fuzzy nonlinear equations can be found in various fields of pure and applied mathematics and engineering.

Previously, Asady *et al.* [1, 2] proposed an iterative technique for solving fuzzy polynomials using Newton's method by reducing them down to a

set of parametric equations, however this method only works if the initial guess is close to the exact solution. Moreover, if convergence does not occur quickly, iterative techniques are problematic as they unnecessarily expand the support of the fuzzy set resulting in vague approximations. At a later time, Abbasbandy *et al.* explored a steepest descent method using parametric equations [3], which produces similar problems as previously mentioned. Abbasbandy *et al.* also proposed using neural networks to solve fuzzy polynomials [4] and system of fuzzy polynomial equations [5], but they consider the unknown variable x to be a crisp quantity. Boukezzoula and Marteau [6] proposed a theoretical method of solving fuzzy quadratic equations in the form $\tilde{a}\tilde{x}^2+\tilde{b}\tilde{x}=\tilde{c}$. However, their method can only be applied when $\tilde{a}$ and $\tilde{c}$ have the same sign and under the assumption that the unknown variable $\tilde{x}$ is either positive or negative and does not account for the case when the membership function of $\tilde{x}$, or any fuzzy quantity in the problem, crosses 0. Moreover, their method is limited to second order fuzzy nonlinear equations and cannot be extended to higher orders, which greatly limits the applicability of the algorithm. Few years later, Zhou and Gan [7] proposed a Hybrid Conjugate Gradient Method for solving fuzzy non-linear equations of any order. Similar to [6], the fuzzy equations are reduced to a set of crisp nonlinear equations under the assumption that the unknown variable, $\tilde{x}$, is either positive or negative. Although this algorithm can be extended to higher orders, it only works when all coefficients are positive. This algorithm also does not consider the case when $\tilde{x}$ contains 0 in its support.

Unlike the aforementioned methodologies, the algorithm we propose can be used to solve a variety of type-1 and interval type-2 fuzzy nonlinear equations of any order, without any type of restriction on the coefficients or the solution. Moreover, the proposed method also works when the support of any fuzzy set in the problem extends across 0.

The paper is organized as follows. In Sec. 2 we briefly discuss some preliminaries in order to clarify the terminology and notations used in the latter sections. Sec. 3 will include discussions on PSO and specifics on how its applied in the given context. Two relevant applications are presented in Sec. 4, and conclusions are drawn in Sec. 5.

2 Preliminaries

A fuzzy set $\tilde{a}$ defined in the universe of discourse X is an ordered pair $\tilde{a}=\{(x,\mu_{\tilde{a}}(x)\mid x\in X)\}$. Each value in the set $x\in X$, has a corresponding membership value $\mu_{\tilde{a}}(x)\ \rightarrow[0,1]$. The elements with a membership value of 0 are considered to be not included in the set, while the elements with a membership value of 1 are considered to be fully included. Similarly, the elements with a membership value between 0 and 1 are considered to be fuzzy members.

In this paper, the tilde symbol placed above a lowercase letter is used to denote a fuzzy set, e.g. $\tilde{x}$. Type-1 fuzzy quantities with a triangular membership function are denoted by t^m placed in-front-of a set of three values enclosed in parenthesis, e.g. $t^1(8,10,11)$. The three values represent the lower-bound, center, and upper-bound values of the fuzzy set respectively, where the superscript value, m, on the letter t represents the membership value of the center element. It is implied that the lower and upper bounds always have a membership value of 0. Similarly, interval type-2 fuzzy quantities are denoted by the type-1 representation of the upper and lower membership functions enclosed in curly brackets e.g. $\left\{\begin{array}{c} t^1(1,2,3)^u \\ t^{0.8}(1.1,2,2.9)^l \end{array}\right\}$.

In this paper, the alpha cut method is used to reduce the fuzzy problem into a number of interval problems which are then sequentially solved using the proposed method. As stated by Zimmerman in [8], "The crisp set of elements that belong to the fuzzy set $\tilde{a}$ at least to the degree α is called the α-level set (or simply the α-cut) $\mathbf{a}_\alpha=\{x\in X:\mu_{\tilde{a}}(x)\geq\alpha\}$ where $\alpha\in[0,1]$". The α-cut operation carried out on a convex fuzzy set results in an interval.

For example, for each $\alpha\in[0,1]$, the fuzzy nonlinear equations given in (1) and (2) will reduce to an interval nonlinear equation in the form of (3) and (4).

$$a_\propto {x_\propto}^n+b_\propto {x_\propto}^{n-1}\cdots+c_\propto x_\propto+d_\propto=z_\propto, \qquad (3)$$

$$a_\propto\exp(x_\propto+b_\propto)=z_\propto. \qquad (4)$$

Where $\mathbf{a}_\propto$, $\mathbf{b}_\propto$, $\mathbf{c}_\propto$, $\mathbf{d}_\propto$, and $\mathbf{z}_\propto$ are known interval quantities derived from the respective fuzzy quantities and $\mathbf{x}_\propto$ is the unknown interval variable. Our methodology is to solve several sets of interval

equations resulting from α-cuts and then to construct the membership function of the fuzzy solution using the interval results.

3 Particle Swarm Optimization (PSO)

3.1 Background

Originally introduced in 1995 by Kennedy and Eberhard, PSO is a search algorithm designed to simulate the social behavior of flocks of birds and schools of fish [9]. In some ways, PSO is similar to evolutionary computation techniques like Genetic Algorithms, but does not incorporate any evolution operators. Here, the particles, initiated randomly in the multidimensional search space, are allowed to fly around while being influenced by the best known optimums.

Generally, at every iteration, t, each particle, $x_i^{(t)} = [x_{i1}^{(t)}, x_{i2}^{(t)}, \cdots, x_{id}^{(t)}]$, is assigned a velocity, $v_i^{(t)} = [v_{i1}^{(t)}, v_{i1}^{(t)}, \cdots, v_{id}^{(t)}]$, according to its current position and the best known positions in the d-dimensional space. The velocity of the given particle for the succeeding iteration, $v_i^{(t+1)} = [v_{i1}^{(t+1)}, v_{i2}^{(t+1)}, \cdots, v_{id}^{(t+1)}]$, is a function of its current velocity, $v_i^{(t)}$, current position, $x_i^{(t)}$, its best known position, $pbest_i = [pbest_{i1}, pbest_{i2}, \cdots, pbest_{id}]$, and the position of the best particle amongst all particles in the group, $gbest = [gbest_1, gbest_2, \cdots, gbest_d]$.

$$\begin{aligned} v_{id}^{(t+1)} =& w \cdot v_{id}^{(t)} + \\ & c_1 \cdot r_1 \cdot \left(pbest_{id} - x_{id}^{(t)} \right) + \\ & c_2 \cdot r_2 \cdot (gbest_d - x_{id}^{(t)}), \end{aligned} \tag{5}$$

$$x_{id}^{(t+1)} = x_{id}^{(t)} + v_{id}^{(t+1)} \tag{6}$$

$i = 1, 2, \cdots, n \quad d = 1, 2, \cdots, m,$

where n is the total number of particles, m is the total number of dimensions, t is the pointer of the current iteration, w is the inertia weight factor, $c1$ and $c2$ are acceleration constants commonly set equal to 2.0, r_1 and r_2 are randomly generated values with uniform distribution in [0,1], $v_i^{(t)}$ is the velocity of particle i during iteration t, $x_i^{(t)}$ is the position of particle i during iteration t.

In general, the inertia weight factor, w, is initially set to 0.9 and decreased linearly to 0.4 according to (7).

$$w = 0.9 - \frac{0.9 - 0.4}{iter_{max}} \times t \tag{7}$$

At each iteration, the current positions of the particles are evaluated using the evaluation function, $f(x)$, if the evaluation value of the given particle's current position, $f(x_i^{(t)})$, is less than or equal to its previously known best position, $f(pbest_i)$, then $pbest_i$ is set equal to x_i^t. If the best $f(pbest_i)$ is less than or equal to the previously known $f(gbest)$, then $gbest$ is set equal to the best $pbest_i$. This process is repeated until $f\left(x_i^{(t)}\right) = 0$ or until $t = iter_{max}$. The optimum point is said to be found if convergence occurs or if a point where $f\left(x_i^{(t)}\right) = 0$ is located.

3.2 PSO for Fuzzy Polynomial Equations

In our case, once the fuzzy nonlinear equation is reduced to an interval nonlinear equation at each $\propto$-cut, the unknown fuzzy variable $\tilde{x}$ is also consequently reduced to an interval $\mathbf{x}_\propto = [\underline{x}, \overline{x}]$.

Since we are dealing with a nonlinear equation, existence of multiple solutions should be acknowledged. In the case of intervals, it is necessary to consider three cases.

Case 1: $\underline{x} \leq 0$ and $\overline{x} \leq 0$

Case 2: $\underline{x} \leq 0$ and $\overline{x} \geq 0$

Case 3: $\underline{x} \geq 0$ and $\overline{x} \geq 0$

Therefore, in order to explore all possible solutions, it is necessary to have three independent groups of particles, each constrained to the search area of one of the aforementioned cases respectively. Particles belonging to any given group are allowed to explore within their respective search area while only being influenced by other members of the same group. However, it is also important to consider the application while solving these problems. For instance, if the application only requires a positive solution then only one group, exploring the case 3 region, will be needed.

At each iteration, the position of each particle, $\mathbf{x}_i^{(t)} = [x_{i1}^{(t)}, x_{i2}^{(t)}]$, where $\underline{x} = \min(x_{i1}^{(t)}, x_{i2}^{(t)})$ and $\overline{x} = \max(x_{i1}^{(t)}, x_{i2}^{(t)})$, is evaluated using (8).

$$f\left(\mathbf{x}_i^{(t)}\right) = \left|\underline{z} - \underline{Z_i}\right| + \left|\overline{z} - \overline{Z_i}\right|, \tag{8}$$

where interval $\mathbf{z}_\propto = [\underline{z}, \overline{z}]$ is given, and the interval $\mathbf{Z}_i = [\underline{Z_i}, \overline{Z_i}]$ is obtained by substituting the current position of the given particle, $\mathbf{x}_i{}^{(t)} = [x_{i1}{}^{(t)}, x_{i2}{}^{(t)}]$, into the problem equation, i.e. (9) or (10).

$$\begin{aligned} Z_i = & a_\propto (x_i{}^{(t)})^n + b_\propto (x_i{}^{(t)})^{n-1} \cdots \\ & + c_\propto \left(x_i{}^{(t)}\right)^n + d_\propto, \end{aligned} \tag{9}$$

$$Z_i = a_\propto exp(x_i{}^{(t)} + b_\propto). \tag{10}$$

It is worth noting that standard algebraic properties do not always extend well to intervals, therefore we strictly adhere to Moore's native form [10] and the theoretical results from our previous work [11] while evaluating these interval functions.

As mentioned in Sec.3.1 $f\left(\mathbf{x}_i^{(t)}\right)$ at each iteration is compared to the previously known $f(pbest_i)$ and $f(gbest)$. Then the velocity and position of the given particle for the succeeding iteration is computed using (5) and (6). This process is repeated until $f\left(\mathbf{x}_i^{(t)}\right) = 0$ is achieved for all three groups, or if $t = iter_{max}$. When $f\left(\mathbf{x}_i^{(t)}\right) = 0$ is achieved, then the solution is said to be found. If convergence occurs at a point where $f\left(\mathbf{x}_i^{(t)}\right) \neq 0$ then the given point can be taken as an approximation, but the existence of the exact solution in the given region cannot be proved or disproved. If convergence does not occur, then two possibilities must be considered.

First, $iter_{max}$ may not be sufficient and should be increased and search parameters may have to be tuned; or it is very likely that multiple approximations resulting in the same evaluation value exists in the given search area but not an exact solution that satisfies the problem. For better understanding, a sequential illustration of the proposed methodology can be seen on Figure 1.

In order to better demonstrate our methodology, here we solve the following second order fuzzy polynomial equation with type-1 triangular coefficients.

$$t^1(10,15,18)\tilde{x}^2 - t^1(5,5.4,6)\tilde{x} = t^1(66,185.5,305)$$

After alpha cuts, the given equation is reduced to a number of interval equations in the following form.

$$a_\propto x_\propto^2 - b_\propto x_\propto = c_\propto$$

Where the coefficients of $\mathbf{a}_\propto$, $\mathbf{b}_\propto$, and $\mathbf{c}_\propto$ are intervals resulting from each $\propto$-cut as shown in Table 1 and $\mathbf{x}_\propto$ is the interval solution at each $\propto$-cut which will later be used to construct the membership function of the fuzzy solution.

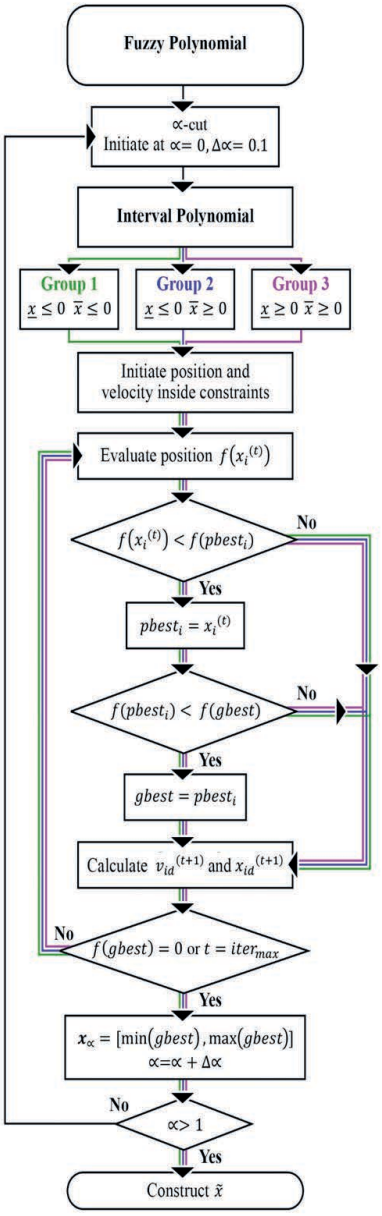

Figure 1. Sequential representation of the proposed methodology

Table 1. Interval Coefficients at Each $\propto$-Cut

$\propto$	$\mathbf{a}_{\propto}$	$\mathbf{b}_{\propto}$	$\mathbf{c}_{\propto}$
0	[10,18]	[5,6]	[66,305]
0.1	[10.5,17.7]	[5.04,5.94]	[77.95,293.05]
0.2	[11,17.4]	[5.08,5.88]	[89.9,281.1]
0.3	[11.5,17.1]	[5.12,5.82]	[101.85,269.15]
0.4	[12,16.8]	[5.16,5.76]	[113.8,257.2]
0.5	[12.5,16.5]	[5.2,5.7]	[125.75,245.25]
0.6	[13,16.2]	[5.24,5.64]	[137.7,233.3]
0.7	[13.5,15.9]	[5.28,5.58]	[149.65,221.35]
0.8	[14,15.6]	[5.32,5.52]	[161.6,209.4]
0.9	[14.5,15.3]	[5.36,5.46]	[173.55,197.45]
1	[15,15]	[5.4,5.4]	[185.5,185.5]

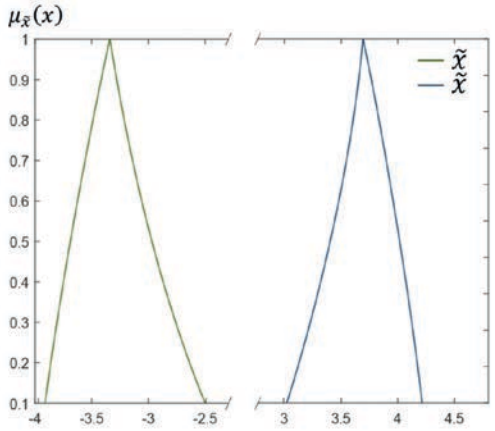

Figure 2. Fuzzy membership function of $\tilde{x}$

Solving the given problem using the proposed methodology produces two fuzzy solutions as shown in Figure 2.

4 Case Studies

Example 1: In the field of business economics, it is common to evaluate the production cost function and demand equation for any given product in order to determine its profit margin. Here, let us assume that the demand equation and cost function for an arbitrary product are $\tilde{s}$ and $\tilde{c}$ respectively.

$$\tilde{s} = \left\{ \begin{array}{c} t^1\,(30.98, 40.14, 40.27)^u \\ t^{0.8}\,(40.06.40.14, 40.2)^l \end{array} \right\}$$

$$- \left\{ \begin{array}{c} t^1(0.000563, 0.000566, 0.000567)^u \\ t^{0.8}(0.000564, 0.000566, 0.0005665)^l \end{array} \right\} \tilde{x}$$

$$\tilde{c} = \left\{ \begin{array}{c} t^1(15.23, 15.25, 15.27)^u \\ t^{0.8}(15.235, 15.25, 15.26)^l \end{array} \right\} \tilde{x}$$

$$- \left\{ \begin{array}{c} t^1(103000, 125000, 148000)^u \\ t^{0.8}(103500, 125000, 147200)^l \end{array} \right\}$$

Where $\tilde{s}$ is the unit price, $\tilde{x}$ is the number of units, and $\tilde{c}$ is the total production cost. The total profit $\tilde{p}$ can therefore be computed using the following equation.

$$\tilde{p} = \tilde{s}\tilde{x} - \tilde{c} \tag{11}$$

Let us assume that the targeted profit for the given product is

$$\left\{ \begin{array}{c} t^1(96000, 100000, 103500)^u \\ t^{0.8}(96800, 100000, 103000)^l \end{array} \right\} \$.$$

Substituting the given quantities into (11) produces the following objective function.

$$\left\{ \begin{array}{c} t^1(96000, 100000, 103500)^u \\ t^{0.8}(96800, 100000, 103000)^l \end{array} \right\} =$$

$$- \left\{ \begin{array}{c} t^1\,(0.000563, 0.000566, 0.000567)^u \\ t^{0.8}\,(0.000564, 0.000566, 0.0005665)^l \end{array} \right\} \tilde{x}^2$$

$$+ \left\{ \begin{array}{c} t^1\,(24.71, 24.89, 25.04)^u \\ t^{0.8}\,(24.8, 24.89, 24.965)^l \end{array} \right\} \tilde{x}$$

$$- \left\{ \begin{array}{c} t^1(103000, 125000, 148000)^u \\ t^{0.8}(103500, 125000, 147200)^l \end{array} \right\}$$

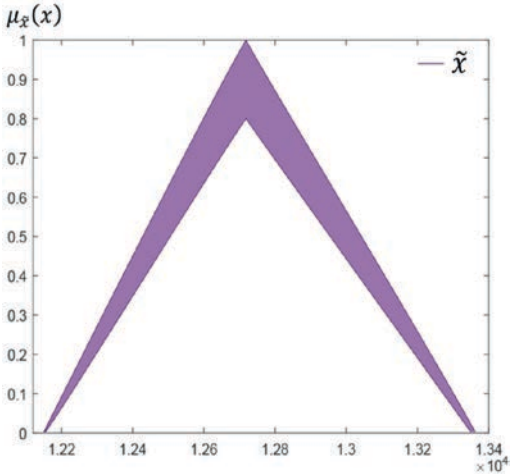

Figure 3. The optimum number of units produced, $\tilde{x}$, in order to meet the targeted profit

Since we are only interested in the positive solution to this problem, we only need one group of particles, constrained in the search area of case 3. Solving the given equation using the proposed methodology results in the following membership function for $\tilde{x}$, which, in the given context, represents the optimum number of units that needs to be produced in order to meet the targeted profit.

Example 2: Let us now consider a more complex application from agricultural engineering. Evapotranspiration (ET) is the process of water lost to the atmosphere through a combination of evaporation and transpiration through plants and soil. Today, because of water-saving initiatives, ET is an important phenomenon that is extensively studied for agricultural applications [12]. The most popular equation used to estimate reference evapotranspiration (ET_o) is called the Penman-Monteith equation [14] given in (12). ET_o measures the rate at which water would be removed from soil and plants from a reference crop (e.g. grass or alfalfa).

$$\widetilde{ET_o} = \frac{0.408\widetilde{\Delta}\left(\widetilde{R_n} - G\right) + \gamma\frac{900}{\tilde{T}+273}\widetilde{u_2}(\widetilde{e_s} - \widetilde{e_a})}{\widetilde{\triangle} + \gamma(1 + 0.34\widetilde{u_2})} \tag{12}$$

Where $\widetilde{\Delta}$ is the slope of saturation vapor pressure curve in $\frac{kPa}{^\circ C}$ (13), $\widetilde{R_n}$ is the mean solar radiation in $\frac{MJ}{m^2}$, $G = 0.082\ \frac{MJ}{m^2}$ is the solar constatnt, γ is the psychometric constant in $\frac{kPa}{^\circ C}$,, $\tilde{T}$ is the average air temperature in ℃, $\widetilde{u_2}$ is the average daily wind speed in $\frac{m}{s}$, RH_{avg} is the average relative humidity, and e_s and e_a are mean saturation vapor pressure (14) and actual vapor pressure (15) in kPa respectively.

$$\widetilde{\Delta} = \frac{4098\left(0.6208\exp\left(\frac{17.27\ \tilde{T}}{\tilde{T}+237.3}\right)\right)}{\left(\tilde{T}+237.3\right)^2}, \tag{13}$$

$$\widetilde{e_s} = 0.6208\exp\left(\frac{17.27\ \tilde{T}}{\tilde{T}+237.3}\right), \tag{14}$$

$$\widetilde{e_a} = \widetilde{e_s}\left(\frac{RH_{avg}}{100}\right). \tag{15}$$

For a greenhouse, it is common to evaluate Eq. (12) to determine the optimum temperature that needs to be maintained in order to stay within the desired ET levels.

Let us assume that

$$\widetilde{ET_o} = t^1(5.33, 9.4, 15.13)\frac{mm}{d},$$

$$\widetilde{R_n} = t^1(25.3, 26.1, 27.5)\frac{MJ}{m^2},$$

$$\gamma = 0.07\frac{kPa}{^\circ C}, \widetilde{u_2} = t^1(0.4, 1, 2.1)\frac{m}{s^2},$$

and $RH_{avg} = 45.58\%$.

Substituting the given values into (12) produces.

$$\widetilde{ET_o} = \frac{t^1(10.29, 11.19)\widetilde{\Delta} + \frac{t^1(13.71, 34.28, 71.99)e_s}{\tilde{T}+273}}{\widetilde{\Delta} + t^1(0.08, 0.09, 0.12)}.$$

It should be noted that both $\widetilde{\Delta}$ and $\widetilde{e_s}$ are exponential functions of $\tilde{T}$, therefore the aforementioned $\widetilde{ET_o}$ equation effectively has exponentials both on the numerator and denominator. However, no changes to the algorithm are needed to solve these types of problems. Solving the given equation for $\tilde{T}$ using the proposed methodology shown in 0 produces the solution shown in Figure 4.

It can be seen that the nonlinearity of the equations resulted in a unique membership function for the temperature. This is a good example to show the inability of simplified fuzzy solutions in capturing the actual solution.

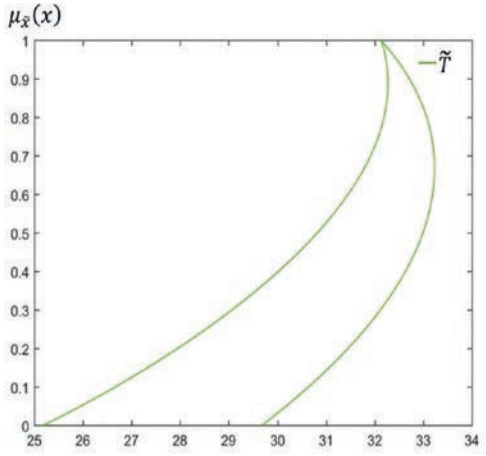

Figure 4. The optimum temperature, $\tilde{T}$, in ℃ that needs to be maintainted to meet the desired ET_o levels

5 Conclusion

In this paper, we proposed a modified Particle Swarm Optimization method which can be used to solve a variety of type-1 and interval type-2 fuzzy nonlinear equations. We better demonstrated the methodology by solving a numerical example and presented two relevant case studies to show how the proposed methods can be applied in a variety of fields. Our future work will include developing a theoretical approach for solving these types of nonlinear equations and applying the proposed methodology for multivariable cases such as systems of fuzzy nonlinear equations.

Acknowledgement

This material is based upon work supported by, or in part by, the U.S. Army Research Laboratory and the U.S. Army Research Office under contract/grant number W911NF-15-1-0498. The authors would like to thank the U.S. Department of Defense for their support.

References

[1] S. Abbasbandy, B. Asady, Newton's method for solving fuzzy nonlinear equations, Applied Mathematics and Computation, Volume 159, Issue 2, Pages 349-356, ISSN 0096-3003.

[2] M. Tavassoli Kajani, B. Asady, A. Hadi Vencheh, An iterative method for solving dual fuzzy nonlinear equations, Applied Mathematics and Computation, Volume 167, Issue 1, Pages 316-323, ISSN 0096-3003.

[3] S. Abbasbandy, A. Jafarian, Steepest descent method for solving fuzzy nonlinear equations, Applied Mathematics and Computation, Volume 174, Issue 1, Pages 669-675.

[4] S. Abbasbandy, M. Otadi, Numerical solution of fuzzy polynomials by fuzzy neural network, Applied Mathematics and Computation, Volume 181, Issue 2, Pages 1084-1089, ISSN 0096-3003.

[5] S. Abbasbandy, M. Otadi, M. Mosleh, Numerical solution of a system of fuzzy polynomials by fuzzy neural network, Information Sciences, Volume 178, Issue 8, Pages 1948-1960, ISSN 0020-0255.

[6] R. Boukezzoula and S. Marteau, Exact solving of the fuzzy equation: $A.X^2 + B.X = C$, The 14th IEEE International Conference on Fuzzy Systems, FUZZ '05., Reno, NV, 2005, pp. 1110-1115.

[7] G. Zhou and Y. Gan, Hybrid conjugate gradient method for solving fuzzy nonlinear equations, Logistics Systems and Intelligent Management, 2010 International Conference on, Harbin, 2010, pp. 462-464.

[8] Zimmerman, H. J. (1996). Fuzzy Set theory and its Application (second edition), Allied Publishers, Indian Reprint.

[9] J. Kennedy and R. Eberhart, Particle swarm optimization, Proc. IEEE Int. Conf. Neural Networks, vol. IV, pp. 1942-1948, 1995.

[10] R.E. Moore, Interval analysis, Prentice-Hall, Englewood Cliffs, NJ, 1966.

[11] S. Sadiqbatcha, and S. Jafarzadeh, An Analytical Approach to Solving Type-1 and Type-2 Fully Fuzzy Linear Systems of Equations, IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Vancouver, 2016.

[12] B. Wu, W. Zhu, N. Yan, X. Feng, Q. Xing and Q. Zhuang, An Improved Method for Deriving Daily Evapotranspiration Estimates From Satellite Estimates on Cloud-Free Days, in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 9, no. 4, pp. 1323-1330, 2016.

[13] K. Djaman, A. B. Balde, A. Sow, B. Muller, S. Irmak, M. K. N'Diaye, B. Manneh, Y. D. Moukoumbi, K. Futakuchi, K. Saito, Evaluation of sixteen reference evapotranspiration methods under sahelian conditions in the Senegal River Valley, Journal of Hydrology: Regional Studies, vol 3, pp. 139-159, ISSN 2214-5818.

[14] Allen, R.G., L.S. Pereira, D. Raes, and M. Smith. 1998. Crop evapotranspiration. Guidelines for computing crop water requirements. FAO Irrigation and Drainage Paper No. 56. United Nations Food and Agricultural Organization, Rome.

Sheriff recently received his B.S. in Computer Engineering from the California State University, Bakersfield. He will soon start his graduate program in Electrical Engineering at the University of California, Riverside. His research interests include fuzzy sets and systems, VLSI systems and computation, and reliability analysis. His current research focus is on modeling and analysis of intermittent and uncertain parameters introduced by large scale renewable energy penetration into the existing power network.

Dr. **Saeed Jafarzadeh**, is an Assistant Professor at the Department of Computer and Electrical Engineering at the California State University, Bakersfield. He received his PhD from the University of Nevada, Reno. His research interests include power systems, smart grid, renewable energy systems, energy markets, and energy conversion. His current research focus is on interdisciplinary aspects of energy industry such as stability analysis of energy markets, forecasting renewable energy resources, quantification of smart grid policies, optimization of energy conversion systems, and power systems dynamic state estimation.

Dr. **Yiannis Ampatzidis** is an assistant professor in the Department of Physics and Engineering at the California State University, Bakersfield. He received his PhD in Agricultural Engineering from the Aristotle University of Thessaloniki in 2010, his MS in Agricultural Engineering in 2005, his B.S. in Agricultural Engineering 2008 and his B.S. in Agriculture and Ecology in 2002 from the same university. His current research focus is on mechatronics, precision agriculture, UAVs and machine systems with special interest in development, implementation and evaluation of agricultural machines and control systems for high value crops.